NASA TECHNICAL MEMORANDUM

NASA TM X-1882

NASA TM X-1882

CHEBYSHEV SERIES SOLUTION OF NONLINEAR ORDINARY DIFFERENTIAL EQUATIONS: INITIAL-VALUE PROBLEMS

by Kin L. Lee and Paul F. Byrd

Ames Research Center
Moffett Field, Calif.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • SEPTEMBER 1969

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM X-1883		

4. Title and Subtitle	5. Report Date
CHEBYSHEV SERIES SOLUTION OF NONLINEAR ORDINARY DIFFERENTIAL EQUATIONS: INITIAL-VALUE PROBLEMS	September 1969
	6. Performing Organization Code
7. Author(s) Kin L. Lee and Paul F. Byrd	8. Performing Organization Report No. A-3302
9. Performing Organization Name and Address NASA Ames Research Center Moffett Field, California 94035	10. Work Unit No. 129-04-04-02-00-21
	11. Contract or Grant No.
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington D. C. 20546	13. Type of Report and Period Covered Technical Memorandum
	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

The approximate Chebyshev series solution of nonlinear ordinary differential equations based on Picard iteration is discussed. Detailed algorithms are provided for the numerical solution of initial-value problems involving (a) a system of n first-order nonlinear differential equations, and (b) an nth-order nonlinear differential equation. Methods to accelerate the convergence of the iterative procedures are also proposed. FORTRAN IV subroutines for the algorithms with options for accelerating convergence are given.

17. Key Words Suggested by Authors	18. Distribution Statement
Near Best Polynomial Approximation Chebyshev Series Solution of Nonlinear Ordinary Differential Equations Picard Iteration Acceleration of Convergence	UNCLASSIFIED - UNLIMITED

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
UNCLASSIFIED	UNCLASSIFIED	75	$3.00

*For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151

TABLE OF CONTENTS

Page

SUMMARY 1
INTRODUCTION 1
SYMBOLS 2
PRELIMINARY ANALYSIS 4
  Chebyshev Series Integration of the First-Order Differential Equation 4
  Accuracy of Solution 8
  Example 9
INTEGRATION OF A SYSTEM OF n FIRST-ORDER DIFFERENTIAL EQUATIONS 11
  Algorithm I (for a system of n first-order equations) 13
  Example I 16
  Remarks on Convergence 20
  Acceleration of Convergence 21
    Steffensen's sequences 21
    Modification of individual entries 25
    Modification of the total array 25
    Modification of rows (MR) 25
    Modification of columns (MC) 26
  An Alternate Procedure 28
INTEGRATION OF AN nth-ORDER DIFFERENTIAL EQUATION 28
  Algorithm II (for an nth-order equation) 30
  Example II 32
  Remarks on Convergence 33
  Acceleration of Convergence 34
APPENDIX A - CHEBYSHEV POLYNOMIALS 36
  Some Properties of Chebyshev Polynomials 36
  Summation of Series of Chebyshev Polynomials 38
APPENDIX B - THE CHEBYSHEV SERIES AND APPROXIMATION BY MODIFIED INTERPOLATION 43
  The Best Approximating Polynomial and the Chebyshev Series 43
  Approximation by Modified Interpolation 46
    The interpolating polynomial $P_{N+1}(t)$ 47
    The modified interpolating polynomial $Q_N(t)$ 49
APPENDIX C - FORTRAN IV SUBROUTINES FOR ALGORITHMS I AND II 53
  Subroutine AL ALG1 53
  Subroutine AL ALG2 56
  Subroutine AL CHBY 59
  Subroutine AL DPIN 60
  Program Listings 62
REFERENCES 74

CHEBYSHEV SERIES SOLUTION OF NONLINEAR ORDINARY DIFFERENTIAL EQUATIONS: INITIAL-VALUE PROBLEMS

By Kin L. Lee and Paul F. Byrd

Ames Research Center

SUMMARY

The approximate Chebyshev series solution of nonlinear ordinary differential equations based on Picard iteration is discussed. Detailed algorithms are provided for the numerical solution of initial-value problems involving (a) a system of n first-order nonlinear differential equations, and (b) an nth-order nonlinear differential equation. Methods to accelerate the convergence of the iterative procedures are also proposed. FORTRAN IV subroutines for the algorithms with options for accelerating convergence are given.

INTRODUCTION

Chebyshev series have been used for obtaining efficient numerical solution to various problems, particularly in approximating functions and in solving certain linear differential equations and Fredholm integral equations (refs. 1 and 2). The usefulness of the Chebyshev series lies in the fact that it very nearly satisfies the condition for optimal polynomial approximation.

The explicit representation of the solution to a nonlinear ordinary differential equation by some economical approximation such as a Chebyshev series is particularly desirable in practice if the solution is part of a larger computation and is repeatedly required for the generation of other relevant information. Besides providing an explicit representation of the solution over the relevant range of the independent variable, the Chebyshev series solution of ordinary initial-value problems has other desirable features that may make it preferable to discrete variable methods (i.e., Runge Kutta and various predictor-corrector procedures). First, the maximum error over the entire interval of integration can be readily and closely estimated by inspection of the coefficients. Second, since the integration can usually be effected over the entire interval or a small number of subintervals, the chances for propagation of round-off error is small. The degree M of the approximating polynomial as well as the interval of integration may be varied as subroutine parameters to obtain any degree of accuracy. These features are lacking in most discrete variable methods. A clear disadvantage of the algorithms for Chebyshev series integration of nonlinear differential equations is the time-consuming feature of their interative construction. However, in the case of a single nth-order differential equation, the application of the Chebyshev series method in conjunction with the method of accelerating convergence may compare favorably with discrete variable methods in computing time.

Approximate Chebyshev series solution for nonlinear differential equations was first proposed by Clenshaw and Norton (ref. 3). The approximate solution, found by Picard iteration, is applied to single first and second-order differential equations. In a later paper (ref. 4), Norton proposed Chebyshev procedures based on Newton iteration to solve the same equations.

The principal objective of this report is to give detailed algorithms based on Picard iteration for obtaining numerical solutions, in the form of an approximate Chebyshev series, of initial-value problems involving (a) a system of n first-order nonlinear differential equations, and (b) an nth-order nonlinear differential equation. These algorithms have been programmed as FORTRAN IV subroutines. The documentation of the subroutines is provided in appendix C.

A basic difference between algorithms presented here and those of Clenshaw and Norton is the choice of the interpolating polynomial. Our preference is based on the discussion of error given in appendix B where certain basic ideas and tools of polynomial approximation are discussed.

Although Chebyshev procedures based on Newton iteration usually seem to converge faster than those based on Picard, they are not easily extended as general algorithms to solve higher order or coupled differential equations. For this reason, no discussion on Newton's method is given here. To compensate for the slower convergence of Picard iteration, two methods are proposed in this report to accelerate the convergence of the above algorithms. The two methods are included as options in the subroutines mentioned earlier.

A discussion on convergence via an example is given to provide insight into the behavior and application of the algorithms of this report.

A summary is given in appendix A on several fundamental properties and tools of the Chebyshev polynomial.

SYMBOLS

A_k	Chebyshev coefficients of the function $\phi(t)$ (see eq. (B11))
B_k	coefficients of the interpolating polynomial $P_{N+1}(t)$ (see eqs. (B23) and (B27))
$b_k^{(0)}$, $b_k^{(1)}$	coefficients of the approximating polynomials for $\phi_j(t)$ and $\phi_j'(t)$ (see eq. (14))
$B_k^{(0)}$, $B_k^{(1)}$	coefficients of the approximating polynomials for $\phi_{j+1}(t)$ and $\phi_{j+1}'(t)$ (see eqs. (16) and (20))

$b_{i,k}^{(0)}$, $b_{i,k}^{(1)}$ coefficients of the approximating polynomials for $\phi_{i,j}(t)$ and $\dot{\phi}'_{i,j}(t)$ (see eq. (46))

$B_{i,k}^{(0)}$, $B_{i,k}^{(1)}$ coefficients of the approximating polynomials for $\phi_{i,j+1}(t)$ and $\phi'_{i,j+1}(t)$ (see eq. (47))

$b_k^{(i)}$ coefficients of the approximating polynomial for $\phi_j^{(i)}(t)$ (see eqs. (67) and (68))

$B_k^{(i)}$ coefficients of the approximating polynomial for $\phi_{j+1}^{(i)}(t)$ (see eqs. (67) and (68))

D_n the set of polynomials of maximum degree n (see eq. (B2))

$E_N(f)$ the minimax of $f(x)$ (see eq. (B3))

$P(x)$ a polynomial in x

$P_k(x)$ a polynomial in x of degree k

$S_N(t)$ the first $N+1$ terms of the Chebyshev series of $\phi(t)$ (see eq. (B13))

$T_k(t)$ the Chebyshev polynomial of the first kind of degree k (see eq. (A1))

$\sigma_N(\phi)$ the maximum error of $S_N(t)$ (see eq. (B14))

$\phi_{i,j}^{(p)}(t)$ the jth approximation to $\phi_i^{(p)}(t)$

$\phi_j^{(i)}(t)$ the jth approximation to $\phi^{(i)}(t)$

$$\sum_{k=0}^{n}{}' u_k = \frac{1}{2} u_o + u_1 + \ldots + u_N$$

$$\sum_{k=0}^{n}{}'' u_k = \frac{1}{2} u_o + u_1 + \ldots + u_{N-1} + \frac{1}{2} u_N$$

ε the convergence criterion or prescribed convergence error of algorithms I and II

$\approx$ approximately equal to, as $P_M(t) \approx \phi(t)$

PRELIMINARY ANALYSIS

Chebyshev Series Integration of the First-Order Differential Equation

The basic ideas and tools used in the construction of algorithms for the approximate Chebyshev series integration of nonlinear differential equations can best be discussed and understood via the first-order differential equation

$$\frac{dF}{dx} = f(x,F) \tag{1}$$

having the initial condition

$$F(a) = \eta \tag{2}$$

(Algorithms for a system of first-order differential equations and an nth-order differential equation are presented in the following sections as algorithms I and II in a form suitable for coding by means of ALGOL, FORTRAN or similar computer languages.)

Consider the sequence of functions $\{F_j(x)\}$ generated by a process attributed to Picard:

$$F_{j+1}(x) = \eta + \int_a^x f(s,F_j)ds \qquad (j = 0,1,\ldots) \tag{3}$$

$$F_0(x) \equiv \eta \tag{4}$$

If $f(x,F)$ is continuous and the partial derivative $\partial f/\partial F$ is bounded in a region including the point (a,η), the above sequence of functions is guaranteed by a theorem of Picard to converge to a function $F(x)$ satisfying equations (1) and (2) in a neighborhood $|x - a| \le h$. If this is the case, then without loss of generality we can consider the solution of the differential equation

$$\frac{d\phi}{dt} = \psi(t,\phi) , \quad -1 \le t \le 1 \tag{5}$$

with

$$\phi(-1) = \eta \tag{6}$$

by means of the iterative procedure involving the equations

$$\phi_{j+1}(t) = \eta + \int_{-1}^t \psi(u,\phi_j)du \tag{7}$$

$$\phi_o(t) \equiv \eta \tag{8}$$

where $\phi_j(t)$ converges uniformly on [-1,1] to the solution of equations (5) and (6).

Although $\psi(t,\phi_j)$, for a fixed j, is an explicit function of t, the integral of (7) is difficult to obtain in practice. However, if $\psi[t,\phi_j(t)]$ can be accurately approximated by a polynomial $P(t)$, then $\phi_{j+1}(t)$ can be evaluated by integrating $P(t)$ term by term. From the point of view of efficient computation, the coefficients of such a polynomial should by readily obtainable by a finite algorithm. Also, for a fixed degree M, this polynomial should be the best possible in the sense of least maximum error (defined by eq. (B2)). Since the computation of the best approximating polynomial is, in general, a nonlinear iterative procedure, the use of it as an effective tool in the approximation of $\psi(t,\phi_j)$ must be precluded. Clenshaw (ref. 3) proposed the use of the interpolating polynomial[1]

$$P_M(t) = \sum_{k=0}^{M}{}'' C_k T_k(t) \tag{9}$$

where $T_k(t)$ are Chebyshev polynomials defined by (see also appendix A)

$$T_k(t) = \cos(k \cos^{-1} t), \qquad -1 \le t \le 1 \tag{10}$$

with the points of interpolation

$$t_r = \cos \frac{r\pi}{M} \qquad (r = 0,1, \ldots, M) \tag{11}$$

where $T_M(t)$ has $M + 1$ extrema $T_M(t_r) = (-1)^r$.

Here, however, we make use of $Q_N(t)$, a modified interpolating polynomial, which is formed by truncating the last term of the interpolating polynomial

$$P_{M+1}(t) = \sum_{k=0}^{M+1}{}'' B_k T_k(t) \tag{12}$$

having

$$t_r = \cos \frac{r\pi}{M+1} \qquad (r = 0,1, \ldots, M+1) \tag{13}$$

as the points of interpolation. These are also the M+2 points where $T_{M+1}(t)$ has extrema, $T_{M+1}(t_r) = (-1)^r$. Analysis in appendix B shows that the maximum error for $Q_M(t)$ as an approximating polynomial for sufficiently large

---

[1] A double prime over the summation sign indicates that the first and last terms are to be halved, while a single prime indicates that only the first term is to be halved.

M is one half that of $P_M(t)$. Numerical examples of Frazer and Hart (ref. 5) also show that $Q_M(t)$ closely approximate the best approximating polynomial. For this reason, $Q_M(t)$ is also called a near-best approximating polynomial.

Now suppose we assume that each member of the sequences $\{\phi_j(t)\}$ and $\{\phi_j'(t)\}$ can be accurately represented by polynomials of degrees $M + 1$ and M, respectively. Assume also that at the jth iteration $\phi_j(t)$ and $\phi_j'(t)$ are known and of the form

$$\left.\begin{aligned} \phi_j(t) &= \sum_{k=0}^{M+1}{}' \, b_k^{(0)} T_k(t) \\ &\text{and} \\ \phi_j'(t) &= \sum_{k=0}^{M}{}' \, b_k^{(1)} T_k(t) \end{aligned}\right\} \tag{14}$$

then an approximate chebyshev series solution for equation (5) can be obtained as follows:

We approximate first $\phi_{j+1}'(t) = \psi[t,\phi_j(t)]$ by the near-best approximating polynomial $Q_M(t)$ to obtain

$$\psi[t,\phi_j(t)] \approx \sum_{k=0}^{M}{}' \, B_k^{(1)} T_k(t) \tag{15}$$

where $B_k^{(1)}$, according to equation (B27) is

$$B_k^{(1)} = \frac{2}{M+1} \sum_{r=0}^{M+1}{}'' \, \psi[t_r,\phi_j(t_r)] T_r(t_k) \tag{16}$$

with

$$t_r = \cos \frac{r\pi}{M+1} \tag{17}$$

Since $B_k^{(1)}$ is a linear combination of the form (A22), it can readily be evaluated for a fixed k by means of recurrence formula (A23). Accordingly,

$$\left.\begin{aligned} c_{M+1} &= \frac{1}{2}\, \psi\left[t_{M+1}, \phi_j(t_{M+1})\right] \\ c_M &= \psi\left[t_M, \phi_j(t_M)\right] + 2t_k c_{M+1} \\ c_r &= \psi\left[t_r, \phi_j(t_r)\right] + 2t_k c_{r+1} - c_{r+2} \\ &\quad (r = M - 1, M - 2, \ldots, 1) \end{aligned}\right\} \tag{18}$$

$$B_k^{(1)} = \frac{1}{2}\psi\left[t_o,\phi_j(t_o)\right] + c_1 t_k - c_2 \tag{19}$$

Denote the integral of equation (16) by the (M+1)st degree polynomial

$$\phi_{j+1}(t) \approx \int \sum_{k=0}^{M}{}' B_k^{(1)} T_k(u)du = \sum_{k=0}^{M+1}{}' B_k^{(0)} T_k(t) \tag{20}$$

Upon performing the indicated integration with the aid of equation (A9) and equating coefficients of $T_k(t)$, one obtains

$$B_k^{(0)} = \frac{B_{k-1}^{(1)} - B_{k+1}^{(1)}}{2k} \qquad (k = 1,2, \ . \ . \ .,M) \tag{21}$$

$$B_{M+1}^{(0)} = \frac{B_M^{(1)}}{M+1} \tag{22}$$

(The same results can also be obtained if equation (B19) is applied.) The constant of integration $(1/2)B_o^{(0)}$ remains to be determined. It can be computed if one notes that

$$\phi_{j+1}(-1) = \eta$$

and by equation (A31) that

$$\phi_{j+1}(-1) = \sum_{k=0}^{M+1}{}' (-1)^k B_k^{(0)}$$

Thus

$$B_o^{(0)} = 2\left[\eta - \sum_{k=1}^{M+1} (-1)^k B_k^{(0)}\right] \tag{23}$$

This completes one iteration. If

$$\left| b_k^{(1)} - B_k^{(1)} \right| < \varepsilon \qquad (k = 0,1, \ . \ . \ .,M) \tag{24}$$

where ε is a prescribed convergence error, we are through. Otherwise, replace each $b_k^{(p)}$ by $B_k^{(p)}$ $(p = 0,1)$ and initiate another solution.

The entire iterative process can be started by setting $\phi_o(t) \equiv \eta$, that is, by taking $b_o^{(0)} = 2\eta$ and $b_k^{(0)} = 0$, for $k = 1,2,\ldots,M+1$. A better initial approximation sometimes can be made by utilizing knowledge of $\psi(t,\phi)$, for example, if $\psi(t,\phi)$ involves only the dependent variable ϕ.

Accuracy of Solution

When the condition (24) is met, we are interested in how well the approximate solution satisfies the given differential equation. If the polynomial approximations (15) are substituted into equation (5), one obtains, upon taking absolute values,

$$\left|\sum_{k=0}^{M}{}' b_k^{(1)} T_k(t) - \psi\left[t, \sum_{k=0}^{M+1}{}' b_k^{(0)} T_k(t)\right]\right|$$

$$\leq \left|\sum_{k=0}^{M}{}' b_k^{(1)} T_k(t) - \sum_{k=0}^{M}{}' B_k^{(1)} T_k(t)\right| + \left|\sum_{k=0}^{M}{}' B_k^{(1)} T_k(t) - \psi\left[t, \sum_{k=0}^{M+1}{}' b_k^{(0)} T_k(t)\right]\right|$$

But

$$\left|\sum_{k=0}^{M}{}' b_k^{(1)} T_k(t) - \sum_{k=0}^{M}{}' B_k^{(1)} T_k(t)\right| < (M+1)\varepsilon \tag{25}$$

by condition (24). Let

$$\psi\left[t, \sum_{k=0}^{M+1}{}' b_k^{(0)} T_k(t)\right] = \sum_{k=0}^{\infty}{}' a_k^{(1)} T_k(t)$$

It then follows from equation (B36) that

$$\left|\sum_{k=0}^{M}{}' B_k^{(1)} T_k(t) - \psi\left[t, \sum_{k=0}^{M+1} b_k^{(0)} T_k(t)\right]\right| \leq \left|a_{M+1}^{(1)}\right| + 2\sum_{k=M+2}^{\infty} \left|a_k^{(1)}\right| \tag{26}$$

Consequently, we see by equations (25) and (26) that

$$\left|\sum_{k=0}^{M}{}' b_k^{(0)} T_k(t) - \psi\left[t, \sum_{k=0}^{M+1}{}' b_k^{(0)} T_k(t)\right]\right| < (M + 1)\varepsilon + \left|a_{M+1}^{(1)}\right| + 2\sum_{k=M+2}^{\infty} |a_k| \tag{27}$$

Hence if $\left|b_k^{(1)} - B_k^{(1)}\right| < \varepsilon$ for each k and if M is sufficiently large, the approximate solution $\sum_{k=0}^{M+1}{}' b_k^{(0)} T_k(t)$, and, *a fortiori*, $\sum_{k=0}^{M+1}{}' B_k^{(0)} T_k(t)$, will satisfy the given differential equation with an error close to $(M + 1)\varepsilon$. In practice, we say that M is "sufficiently large" when larger values provide no change greater than ε in the coefficients. Also, since the Chebyshev series is unique, when $(M + 1)\varepsilon$ is made small by an appropriate choice of ε and M, the coefficients $B_k^{(0)}$ of the finite series will closely approximate those of the Chebyshev coefficients $A_k^{(0)}$ of the solution (see eq. (B11).

Example

To illustrate the accuracy of the above procedure, let us find a polynomial approximation for $\tan[(\pi/8)(t + 1)]$ for $-1 \leq t \leq 1$ with a maximum error less than 0.5×10^{-8}. One can easily verify that the given function satisfies the differential equation

$$\frac{d\phi}{dt} = \frac{\pi}{8}[1 + \phi^2(t)], \qquad -1 \leq t \leq 1 \tag{28}$$

with

$$\phi(-1) = 0 \tag{29}$$

Hence, the method given in this section is applicable.

The approximate Chebyshev coefficients for both the solution and its derivative corresponding to $M = 16$ and $\varepsilon = 0.5\times10^{-10}$ are shown in table I(a). A total of 13 iterations were required. Tabulated values of the approximate Chebyshev series corresponding to discrete points of the independent variable are given in table I(b). Numerical results suggest that $M = 16$ is sufficiently large since the coefficients corresponding to $\varepsilon = 0.5\times10^{-10}$ for $M > 16$ yield no change greater than ε. In view of equation (26), the approximate solution must satisfy the differential equation with an error bound close to 8.5×10^{-10}. The same conclusion can be drawn by the examination of the coefficients alone. In fact, since $B_k^{(1)}$ is approximately equal to $a_k^{(1)}$

according to equation (B28) and $\left|B_{k+1}^{(1)}/B_k^{(1)}\right| < 1/5$ for $K \geq 7$, the right member of inequality (27) gives us

$$17\varepsilon + a_{17} + 2\sum_{k=18}^{\infty} |a_k| \approx 8.7\times10^{-10}$$

A check of the tabulated values of table I(b) with those of Abramowitz and Stegun (ref. 6) shows agreement to 10 decimal places.

TABLE I.- CHEBYSHEV SERIES APPROXIMATION OF TAN $\left[\frac{\pi}{8}(t + 1)\right]$, $(-1 \leq t \leq 1)$

(a) Approximate Chebyshev coefficients

k	$B_k^{(0)}$	$B_k^{(1)}$
0	0.9113043408269388D 00	0.1043307398148425D 01
1	0.4894686436450291D 00	0.1835797842596252D 00
2	0.4284834890908355D-01	0.6437011085836632D-01
3	0.1024434335477160D-01	0.1218638862329105D-01
4	0.1453268753787471D-02	0.2904050729736723D-02
5	0.2787802534394446D-03	0.5602385929912773D-03
6	0.4483018228371541D-04	0.1162481953422768D-03
7	0.7992572697332448D-05	0.2227640558669249D-04
8	0.1340847727289307D-05	0.4352177579622525D-05
9	0.2331260728730840D-06	0.8228419500635855D-06
10	0.3968754154824808D-07	0.1559082679070123D-06
11	0.6840670790663591D-08	0.2909111909862381D-07
12	0.1170504525821132D-08	0.5413510512413251D-08
13	0.2011467240884598D-09	0.9990104789166475D-09
14	0.3447781322812264D-10	0.1836956861132969D-09
15	0.5912174018325965D-11	0.3363170852921371D-10
16	0.1050990891537928D-11	0.6330465563517943D-11
17	0.1861901636328807D-12	0.0000000000000000D-38

(b) Function values

t	$\tan\left[\frac{\pi}{8}(t + 1)\right]$
-1.0	0.0000000000000000D-38
-0.8	0.7870170682457329D-01
-0.6	0.1583844403245379D 00
-0.4	0.2400787590801460D 00
-0.2	0.3249196962328421D 00
0.0	0.4142135623731530D 00
0.2	0.5095254494943512D 00
0.4	0.6128007881399821D 00
0.6	0.7265425280053249D 00
0.8	0.8540806854634090D 00
1.0	0.9999999999998522D 00

For economy of computation, note that if $N_1 < N$, then

$$\left| \sum_{k=0}^{N}{}' B_k T_k(t) - \sum_{k=0}^{N_1}{}' B_k T_k(t) \right| \le \sum_{k=N_1+1}^{N} |B_k| \tag{30}$$

(See also eq. (B16)). Thus, in the case of the approximating polynomial of the above example, ignoring the last six terms from the finite series results in a maximum error of 0.14×10^{-8}. Hence, an 11th degree instead of a 17th degree, polynominal can be used to approximate $\tan[(\pi/8)(1 + t)]$ and still satisfy the maximum error requirement of 0.5×10^{-8}.

INTEGRATION OF A SYSTEM OF n FIRST-ORDER DIFFERENTIAL EQUATIONS

In this section, the basic ideas applied to the construction of an approximate Chebyshev series solution for a single first-order differential equation are extended to provide an algorithm for the solution of a system of n first-order differential equations. It is given in sufficient detail to facilitate computer programming as well as the discussion of acceleration of convergence. The basis for the more general algorithm is the following theorem.

Theorem. Let a system of n first-order differential equations be defined by

$$\frac{dF_i}{dx} = f_i(x,F_1,F_2, \ldots ,F_n) , \quad |x - a| \le C_o , \quad (i = 1,2, \ldots ,n) \tag{31}$$

with the initial conditions

$$F_i(a) = \eta_i , \quad (i = 1,2, \ldots ,n) \tag{32}$$

Furthermore, let each of the functions f_i be continuous and have bounded partial derivatives

$$\left|\frac{\partial f_i}{\partial F_j}\right| \le K , \quad (i,j = 1,2, \ldots ,n) \tag{33}$$

in the region

$$|x - a| \le C_o , \quad |F_i - \eta_i| \le C_i , \quad (i = 1,2, \ldots ,n) \tag{34}$$

If

$$h = \min\left(C_o, \frac{C_1}{L} , \frac{C_2}{L} , \ldots \frac{C_n}{L}\right) \tag{35}$$

where

$$L = \max_i \left[\max \left| f_i(x, F_1, F_2, \ldots, F_n) \right| \right] \tag{36}$$

in the region defined by (34), then the sequence of n functions $\left\{F_{1,j}(x), F_{2,j}(x), \ldots, F_{n,j}(x)\right\}_{j=0}^{\infty}$ defined by

$$F_{i,j+1}(x) = \eta_i + \int_a^x f_i(s, F_{1,j}, F_{2,j}, \ldots, F_{n,j})\,ds, \qquad (i = 1, 2, \ldots, n) \tag{37}$$

with

$$F_{i,0}(x) \equiv \eta_i, \qquad (i = 1, 2, \ldots, n) \tag{38}$$

converges uniformily on $|x - a| \leq h$ to a unique set of functions of $F_1(x)$, $F_2(x), \ldots, F_n(x)$ satisfying equations (31) and (32). (For proof of a similar theorem, see Tenenbaum and Pollard (ref. 7)).

Besides providing an iterative procedure (eqs. (37) and (38)) for obtaining a solution, the above theorem also guarantees an interval of convergence. However, the estimate h in equation (35) usually proves to be conservative if not difficult to find. In practice, the interval of convergence is usually assumed or determined by trial and error.

Suppose that the system (31) has a unique solution $F_i(x)$ $(i = 1, 2, \ldots, n)$ on $a \leq x \leq b$ satisfying the initial conditions given by equation (32). In order to construct the Chebyshev series for $F_i(x)$, make the change of independent variable

$$x = ct + d,$$

with

$$c = \frac{b-a}{2} \tag{39}$$

and

$$d = \frac{b+a}{2}$$

so that

$$F_i(x) = \phi_i(t), \qquad \frac{dF_i}{dx} = c^{-1}\phi_i'(t), \qquad F_i(a) = \phi_i(-1) = \eta_i$$

Substitution in equations (31) and (32) then yields

$$\phi_i'(t) = \psi_i(t, \phi_1, \phi_2, \ldots \phi_n), \qquad -1 \leq t \leq 1, \qquad (i = 1, 2, \ldots, n) \tag{40}$$

with

$$\phi_i(-1) = \eta_i, \qquad (i = 1, 2, \ldots, n) \tag{41}$$

The same change of variable for equations (37) and (38) gives the sequence of n functions $\left\{\phi_{1,j},\phi_{2,j}, \ldots,\phi_{n,j}\right\}_{j=0}^{\infty}$ defined by

$$\phi_{i,j+1}(t) = \eta_i + \int_{-1}^{t} \psi_i(u,\phi_{1,j},\phi_{2,j}, \ldots,\phi_{n,j})du , \qquad (i = 1,2, \ldots,n) \tag{42}$$

and

$$\phi_{i,0}(t) \equiv \eta_i , \qquad (i = 1,2, \ldots,n) \tag{43}$$

which converges uniformly on the closed interval [-1,1] to a set of functions $\phi_1(t)$, $\phi_2(t)$, . . ., $\phi_n(t)$ satisfying equations (40 and (41). One also obtains by differentiation the equations

$$\phi'_{i,j+1}(t) = \psi_i(t,\phi_{1,j},\phi_{2,j}, \ldots,\phi_{n,j}) , \qquad (i = 1,2, \ldots,n) \tag{44}$$

with

$$\phi'_{i,1}(t) = \psi(t,\eta_1,\eta_2, \ldots,\eta_n) \tag{45}$$

We are now ready to proceed with the algorithm for an approximate Chebyshev series solution of n first-order differential equations.

Algorithm I

As in the case of a single first-order differential equation, it is assumed that the sequences $\left\{\phi_{1,j},\phi_{2,j}, \ldots,\phi_{n,j}\right\}_{j=0}^{\infty}$ and $\left\{\phi'_{1,j},\phi'_{2,j}, \ldots,\phi'_{n,j}\right\}_{j=0}^{\infty}$ can be accurately approximated by polynomials of degree $M + 1$ and M, respectively. For simplicity of notation, let

$$\phi_{i,j}(t) = {\sum_{k=0}^{M+1}}' b_{k,i}^{(0)}T_k(t) , \qquad \phi'_{i,j}(t) = {\sum_{k=0}^{M}}' b_{k,i}^{(1)}T_k(t) \tag{46}$$

$$\phi_{i,j+1}(t) = {\sum_{k=0}^{M+1}}' B_{k,i}^{(0)}T_k(t) , \qquad \phi'_{i,j+1}(t) = {\sum_{k=0}^{M}}' B_{k,i}^{(1)}T_k(t) \tag{47}$$

Also, let ε be a prescribed convergence error between $b_{k,i}^{(1)}$ and $B_{k,i}^{(1)}$. The approximate solution of (40) can be obtained as follows:

1. Set

$$t_k = \cos\frac{k\pi}{M+1}, \qquad k = 0, 1, \ldots, M+1$$

(These are the M + 2 points where $T_{m+1}(t)$ assumes its extrema.)

2. Set

$$b_{o,i}^{(0)} = 2\eta_i, \qquad (i = 1,2, \ldots, n)$$

$$b_{k,i}^{(0)} = 0, \qquad (i = 1,2, \ldots, n;\ k = 1,2, \ldots, M+1)$$

$$b_{k,i}^{(1)} = 0, \qquad (i = 1,2, \ldots, n;\ k = 0,1, \ldots, M)$$

(This is equivalent to the initial approximations $\phi_i(t) \equiv \eta_i$, $(i = 1,2, \ldots, n)$).

3. Compute

$$\phi_{i,j}(t_k) = \sum_{r=0}^{M+1}{}' b_{r,i}^{(0)} T_k(t_k), \qquad (i = 1,2, \ldots, n;\ k = 0,1, \ldots, M+1)$$

4. Compute

$$\phi'_{i,j+1}(t_k) = \psi_i[t_k, \phi_{1,j}(t_k), \ldots, \phi_{n,j}(t_k)]$$

$$(i = 1,2, \ldots, n;\ k = 0,1, \ldots, M+1)$$

5. Compute (by making use of eq. (B27))

$$B_{k,i}^{(1)} = \frac{2}{M+1} \sum_{r=0}^{M+1}{}'' \phi'_{i,j+1}(t_r) T_r(t_k), \qquad (i = 1,2, \ldots, n;\ k = 0,1, \ldots, M)$$

(We have here approximated $\phi'_{i,j+1}(t)$ by the Mth degree polynomial $\sum_{k=0}^{M}{}' B_k^{(1)} T_k(t)$, where according to Theorem B5

$$B_{k,i}^{(1)} \rightarrow \frac{2}{\pi}\int_{-1}^{1} \psi_i\left[t,\phi_{1,j},\phi_{2,j},\ . \ . \ . \ ,\phi_{n,j}\right]T_k(t)(1 - t^2)^{-1/2}\, dt$$

for a sufficiently large M.)

6. Compute (by eq. (B19))

$$B_{k,i}^{(0)} = \frac{B_{k-1,i}^{(1)} - B_{k+1,i}^{(1)}}{2K}, \qquad (i = 1,2,\ . \ . \ . \ ,n;\ k = 1,2,\ . \ . \ . \ ,M)$$

$$B_{M+1,i}^{(0)} = \frac{B_M^{(1)}}{2(M+1)}, \qquad (i = 1,2,\ . \ . \ . \ ,n)$$

7. Compute (using the fact that $\phi_{i,j+1}(-1) = \eta_i$ and eq. (A31))

$$B_{o,i}^{(0)} = 2\left[\eta_i - \sum_{k=1}^{M+1}(-1)^k B_{k,i}^{(0)}\right], \qquad (i = 1,2,\ . \ . \ . \ ,n)$$

(We have in this and the above step integrated

$$\phi'_{i,j+1}(t) = {\sum_{k=0}^{M}}' B_{k,i}^{(1)}T_k(t)$$

which results in the $(M + 1)$st degree polynomial

$$\phi_{i,j+1}(t) = {\sum_{k=0}^{M+1}}' B_{k,i}^{(0)}T_k(t))$$

8. If $|B_{k,i}^{(1)} - b_{k,i}^{(1)}| < \varepsilon$ for all k and i, we are through. Otherwise,

9. Set

$$b_{k,i}^{(p)} = B_{k,i}^{(p)}, \qquad (p = 0,1)$$

for each k and i and return to step 3.

Note: To take into consideration the case when $\phi'_{i,1}(t) = \phi'_{i,0}(t) \equiv 0$ for each i, step 8 should be bypassed until the second iteration.

Example I

As an example, we solve the boundary-layer equation

$$\frac{d^3F}{dx^3} + 2F\frac{d^2F}{dx} - \left(\frac{dF}{dx}\right)^2 + 1 = 0 , \qquad 0 \le x \le 6 \tag{48}$$

with the initial conditions

$$F(0) = 0, \quad \left.\frac{dF}{dx}\right|_{x=0} = 0, \quad \left.\frac{d^2F}{dx^2}\right|_{x=0} = 1.311937693880 \tag{49}$$

using algorithm I with a prescribed convergence error $\varepsilon = 0.5\times10^{-10}$.

We first rewrite the given differential equation as a system of three first-order differential equations. Let $F_1(x) = F(x)$, $F_2(x) = dF/dx$, and $F_3(x) = d^2F/dx^2$. Equation (48) can then be equivalently written as

$$\frac{dF_1}{dx} = F_2(x), \quad \frac{dF_2}{dx} = F_3(x), \quad \frac{dF_3}{dx} = F_2{}^2(x) - 2F_1(x)F_3(x) - 1$$

with initial conditions $F_1(0) = 0$, $F_2(0) = 0$, $F_3(0) = 1.311937693880$.

We were unable to find a solution to the example by the method of this section over the entire interval because the iteration process failed to converge. However, we can obtain a solution over subintervals of the given interval. Let the given interval be subdivided by the $q + 1$ points

$$0 = x_0 < x_1 < \ldots < x_q = 6 \tag{50}$$

On each of the subintervals $[x_{r-1},x_r]$ $(r = 1,2, \ldots ,q)$ the change of variable $x = ct + d$, $c = (x_r - x_{r-1})/2$, $d = (x_r + x_{r-1})/2$ enables one to rewrite equation (48) as

$$\phi_1'(t) = c\phi_2(t)_2, \quad \phi_2'(t) = c\phi_3(t) , \quad \phi_3'(t) = c\left[\phi_2{}^2(t) - 2\phi_1(t)\phi_3(t) - 1\right]$$

The solution can then be found for one subinterval at a time. Initial conditions for each subinterval are the function values of the end points of the previous subinterval, except the first where $\phi_1(-1) = F_1(0)$, $\phi_2(-1) = F_2(0)$, and $\phi_3(-1) = F_3(0)$. The approximate Chebyshev coefficients $B_{k,1}^{(0)}$ of $F_1(x) = F(x)$ for the case with $M = 11$ and $x_0 = 0$, $x_1 = 1$, $x_2 = 2$, . . . , $x_6 = 6$ are given by table II(a). The coefficients for the first and second derivatives are not tabulated but may be generated in terms of $B_{k,1}^{(0)}$ from equation (B19). A tabulation of the values of $F(x)$, dF/dx, and d^2F/dx^2

corresponding to $x = 0(0.2)6.0$ is provided by table II(b). Numerical results indicate that the approximate solution satisfies the differential equation with an error bound of the order 0.5×10^{-10}, since $B_{k,i}^{(0)}$ exhibits no change to 10 decimal places for $M > 11$ (see Preliminary Analysis). Note that the first and second derivatives approach unity and zero (accurate to 10 decimal places), respectively. This is because the initial value of the second derivative was determined numerically from the solution of the boundary-value problem involving the same differential equation (45) and the boundary values

$$F(0) = 0 \ , \qquad \left.\frac{dF}{dx}\right|_{x=0} = 0 \ , \qquad \left.\frac{dF}{dx}\right|_{x=\infty} = 1$$

(See L. Fox (ref. 8) for the numerical solution of boundary-value problems by means of initial-value techniques).

TABLE II.- APPROXIMATATE CHEBYSHEV SERIES OF EXAMPLE I; $M = 11$ AND $\varepsilon = 0.5\times10^{-10}$

(a) Approximate Chebyshev coefficients of $F(x) = F_1(x)$

k	$B_{k,1}^{(0)}$		k	$B_{k,1}^{(0)}$	
	0.3892366850285570D 00		0	0.5862202712178695D 01	
1	0.2510773451273367D 00		1	0.4999948514648397D 00	
2	0.5149697317862475D-01		2	0.2955377779468674D-05	
3	-0.4884131498391032D-02		3	-0.1245878270158787D-05	
4	0.9147538219192970D-04		4	0.4005516561512526D-06	
5	0.1415676089930017D-04		5	-0.1007087050914716D-06	
6	0.4392922674440762D-06		6	0.2006791017080924D-07	
7	-0.1455950532358585D-06	$0 \leq x \leq 1$	7	-0.3171829281200812D-08	$3 \leq x \leq 4$
8	-0.4849170368489048D-08		8	0.3907806881905951D-09	
9	0.8282513783615847D-09		9	-0.3545559422189501D-10	
10	0.9882967257068829D-10		10	0.1928744432685055D-11	
11	-0.6903978861669351D-11		11	0.2469520799973746D-13	
12	-0.8827151972331154D-12		12	-0.1720633666410817D-13	
0	0.1906417059176552D 01		0	0.7862196443651593D 01	
1	0.4721517772513179D 00		1	0.4999999943549484D 00	
2	0.9310264019825763D-02		2	0.3692652891795766D-08	
3	-0.1833951624942309D-02		3	-0.1878812574654078D-08	
4	0.2059970855772206D-03		4	0.7638849269566333D-09	
5	-0.8457783925547068D-05		5	-0.2536970874119266D-09	
6	-0.8661141518947492D-06		6	0.7001695230546277D-10	
7	0.1097311409086846D-06	$1 \leq x \leq 2$	7	-0.1625416567214393D-10	$4 \leq x \leq 5$
8	0.3074443225322085D-08		8	0.3200698242324569D-11	
9	-0.1144396357769898D-08		9	-0.5361639630217125D-12	
10	0.1743676140975036D-10		10	0.7642312708592647D-13	
11	0.9458996086397773D-11		11	-0.9226205658049735D-14	
12	-0.4752447471050794D-12		12	0.9246076126956381D-15	
0	0.3863311229682918D 01		0	0.9862196437121186D 01	
1	0.4991658557862458D 00		1	0.5000000000008645D 00	
2	0.3890170133408291D-03		2	0.7747691377346653D-12	
3	-0.1226429401112712D-03		3	-0.4426482328826845D-12	
4	0.2723834050810958D-04		4	0.1981233464324698D-12	
5	-0.4276208831696196D-05		5	-0.8628422050923251D-13	
6	0.4483551620850419D-06		6	0.2645800245559826D-13	
7	-0.2359560210099267D-07	$2 \leq x \leq 3$	7	-0.9037083251041076D-14	$5 \leq x \leq 6$
8	-0.1071584776352978D-08		8	0.1891137709394048D-14	
9	0.3086935209765476D-09		9	-0.5728442411781188D-15	
10	-0.1991941542267561D-10		10	0.7540264708912517D-16	
11	-0.7958215315696075D-12		11	-0.2460153293203471D-16	
12	0.2178984230703938D-12		12	0.2698458740408366D-17	

TABLE II.- APPROXIMATE CHEBYSHEV SERIES OF EXAMPLE I; $M = 11$ AND $\varepsilon = 0.5\times10^{-10}$

(b) Values of the approximate solution and the first two derivatives for $x = 0(0.2)6.0$

x	F(x)	$\frac{dF}{dx}$	$\frac{d^2F}{dx^2}$
0.0	0.1387778780781446D-16	0.0000000000000000D-38	0.1311937693880000D 01
0.2	0.2490564860649391D-01	0.2423943538939113D 00	0.1112107122641016D 01
0.4	0.9430258133873859D-01	0.4449866213741513D 00	0.9145465290677766D 00
0.6	0.2003061101255821D 00	0.6087099439315713D 00	0.7245087590634542D 00
0.8	0.3353391239337122D 00	0.7357728939772173D 00	0.5492147378176965D 00
1.0	0.4924144512322781D 00	0.8298680510454096D 00	0.3959361369563590D 00
1.2	0.6654189143037998D 00	0.8959772698742670D 00	0.2699903096575698D 00
1.4	0.8493213259061367D 00	0.9398267343977971D 00	0.1733584569022934D 00
1.6	0.1040250891316518D 01	0.9671738223714611D 00	0.1044301711933930D 00
1.8	0.1235435703128323D 01	0.9831581570755136D 00	0.5884905029559881D-01
2.0	0.1433033404109585D 01	0.9918921621644430D 00	0.3095379362802304D-01
2.2	0.1631909451187109D 01	0.9963447565741975D 00	0.1517036399321181D-01
2.4	0.1831417188863473D 01	0.9984593547253356D 00	0.6918247270753256D-02
2.6	0.2031215672918176D 01	0.9993937528203719D 00	0.2932544923138562D-02
2.8	0.2231138667864316D 01	0.9997775510127270D 00	0.1154412461402519D-02
3.0	0.2431111230808782D 01	0.9999239691430687D 00	0.4217264542212354D-03
3.2	0.2631102124524431D 01	0.9999758156979835D 00	0.1428864892765967D-03
3.4	0.2831099311572934D 01	0.9999928465126406D 00	0.4487654564845384D-04
3.6	0.3031098503447131D 01	0.9999980337347978D 00	0.1305947442842973D-04
3.8	0.3231098287668217D 01	0.9999994980713486D 00	0.3520034178472342D-05
4.0	0.3431098234149990D 01	0.9999998810724133D 00	0.8785038983643211D-06
4.2	0.3631098221826554D 01	0.9999999738584204D 00	0.2029509793444975D-06
4.4	0.3831098219193559D 01	0.9999999946730537D 00	0.4338982542142417D-07
4.6	0.4031098218672120D 01	0.9999999989958596D 00	0.8583529424554144D-08
4.8	0.4231098218576820D 01	0.9999999998268141D 00	0.1572242307020353D-08
5.0	0.4431098218561269D 01	0.9999999999737648D 00	0.2766056552012288D-09
5.2	0.4631098218559299D 01	0.9999999999991659D 00	0.4395167933392727D-10
5.4	0.4831098218559648D 01	0.1000000000003167D 01	0.7029429704709523D-11
5.6	0.5031098218560366D 01	0.1000000000003867D 01	0.1593260600584493D-11
5.8	0.5231098218561158D 01	0.1000000000004142D 01	0.3583023509204903D-12
6.0	0.5431098218561917D 01	0.1000000000005115D 01	-0.8557528262266267D-11

Remarks on Convergence

In the application of the method of the foregoing section the integration involved in each of the iterations

$$\phi_{i,j+1}(t) = \eta_i + \int_{-1}^{t} \psi_i(u,\phi_{1,j},\phi_{2,j}, \ . \ . \ . \ ,\phi_{n,j})du$$

is an approximate one. The accuracy by which $\phi_{i,j+1}(t)$ can be evaluated depends on how accurately $\psi_i(t,\phi_{1,j},\phi_{2,j}, \ . \ . \ . \ ,\phi_{n,i})$ is approximated, or, equivalently, on the degree M of the polynomial $\sum_{k=0}^{M}{}' B_{i,k}^{(1)}T_k(t)$ used to approximate $\psi_i(t,\phi_{1,j}, \ . \ . \ . \ ,\phi_{n,j})$. It is to be expected that the rate of convergence of the functions $\phi_{i,j}(t)$ may be influenced by M. Table III suggests that this is indeed the case. Numerical results showed that an $M \geq 11$ is sufficient to provide an approximate solution to 10 decimal places for example I. However, the number of iterations required, for a fixed convergence criterion ε, may vary with M. Note that the number of iterations needed for the first two intervals remained fixed for each M. On the other hand, the number of iterations required for convergence over the remaining intervals decreases with increasing M. The same table shows, in the case of $5 \leq x \leq 6$, that there is a rather sharp decrease in the number of iterations when M is incremented only by 1.

TABLE III.- THE INFLUENCE OF M ON THE CONVERGENCE OF ALGORITHM I.

M \ Interval	$0 \leq x \leq 1$	$1 \leq x \leq 2$	$2 \leq x \leq 3$	$3 \leq x \leq 4$	$4 \leq x \leq 5$	$5 \leq x \leq 6$
11	17	21	30	38	44	53
12	17	21	28	36	41	42
13	17	21	26	34	39	39
14	17	21	25	32	37	29
15	17	21	25	29	33	28

Each entry of this table indicates the number of iterations required by algorithm I to solve equation (48) having the indicated M and interval of integration. The convergence criterion is $\varepsilon = 0.5\times10^{-10}$. Since the approximate Chebyshev coefficients agree to 10 decimal places for all $M \geq 11$, the solution for each M is also accurate to 10 decimal places.

An observation one can make is that with an appropriate choice of M the computing time used for solving a particular problem can be minimized. When an appropriate choice of M is not available it may be prudent to use a larger M than is deemed necessary for a prescribed accuracy (e.g., using

M = 15 instead of 11 for example I) to insure convergence within a moderate number of iterations. In the next section it will be shown how the convergence of the iterative process can be accelerated for a fixed M, thereby improving the efficiency of algorithm I.

Acceleration of Convergence

To economize computation, two methods consisting of "modification of rows" and "modification of columns" are proposed in this section to accelerate the convergence of the algorithm. Before we proceed further, the basic tool required by these methods must be given.

Steffensen's sequences.- Consider first the modified sequence associated with the single-point scalar iteration function $y_{j+1} = f(y_j)$ having the properties: (a) the equation $y = f(y)$ has a solution $y = u$; (b) the third derivative $f'''(y)$ is continuous in a neighborhood of u with $f'''(u) \neq 1$. Let the sequence be denoted by

$$\begin{array}{llll} y_0 = y_0^{(0)} & y_0^{(1)} & y_0^{(2)} & \cdots \\ y_1^{(0)} & y_1^{(1)} & y_1^{(2)} & \cdots \\ y_2^{(0)} & y_2^{(1)} & y_2^{(2)} & \cdots \end{array} \tag{51}$$

The first member $y_0^{(0)}$ of the sequence is an initial approximation to u. The other members are evaluated in the order indicated by the formulas

$$y_1^{(r)} = f\left(y_0^{(r)}\right), \qquad y_2^{(r)} = f\left(y_1^{(r)}\right) \tag{52}$$

$$y_0^{(r+1)} = \begin{cases} y_2^{(r)} - \dfrac{\left(y_2^{(r)} - y_1^{(r)}\right)^2}{y_2^{(r)} - 2y_1^{(r)} + y_0^{(r)}}, & \text{if } y_2^{(r)} - 2y_1^{(r)} + y_0^{(r)} \neq 0 \\[2ex] y_2^{(r)}, & \text{if } y_2^{(r)} - 2y_1^{(r)} + y_0^{(r)} = 0 \end{cases} \tag{53}$$

It can be shown that the sub-sequence $\left\{y_0^{(r)}\right\}$ having any prescribed y_0 is quadratically convergent (see, e.g., ref. 9). Therefore, the application of equations (52) and (53) would effectively result in accelerating the convergence of the iteration function $y_{j+1} = f(y_j)$ whose regular sequence may be only linearly convergent. Equation (53) is known as Aitken's formula, but the scheme consisting of equations (52) and (53) in the evaluation of the sequence (52) is due to Steffensen (see ref. 9).

Consider next the more general case of the single-point vector iteration $\vec{f}$ defined by

$$\vec{y}_{j+1} = \vec{f}(\vec{y}_j) \tag{54}$$

where $\vec{y}_j$ is the N-dimensional vector

$$\vec{y}_j = (y_{1,j}, y_{2,j}, \ . \ . \ . \ , y_{N,j})^T$$

it is also assumed that the vector equation $\vec{y} = \vec{f}(\vec{y})$ has a solution $\vec{y} = \vec{u}$, and $\vec{f}(\vec{y})$ has certain desirable properties analogous to the scalar function $f(y)$. However, even without knowing precisely what the desirable properties should be, we can, at least formally, generate a sequence of vectors analogous to the sequence associated with the scalar iteration function. This is accomplished by recomputing a new $\vec{y}_0$ after every $N + 1$ sucessive applications of (54).

Let the sequence be denoted by

$$\begin{array}{ccccc} \vec{y}_0 = \vec{y}_0^{(0)} & \vec{y}_0^{(1)} & \vec{y}_0^{(2)} & . \ . \ . \\ \vec{y}_1^{(0)} & \vec{y}_1^{(1)} & \vec{y}_1^{(2)} & . \ . \ . \\ \vec{y}_2^{(0)} & \vec{y}_2^{(1)} & \vec{y}_2^{(2)} & . \ . \ . \\ . & . & . & \\ . & . & . & \\ . & . & . & \\ \vec{y}_{N+1}^{(0)} & \vec{y}_{N+1}^{(1)} & \vec{y}_{N+1}^{(2)} & . \ . \ . \end{array} \tag{55}$$

Define two $N \times N$ matrices ΔY_0 and $\Delta^2 Y_0$ by

$$\Delta Y_0 = \left(\Delta\vec{y}_0^{(r)}, \Delta\vec{y}_1^{(r)}, \ . \ . \ . \ , \Delta\vec{y}_{N-1}^{(r)}\right)$$

$$\Delta^2 Y_0 = \left(\Delta^2\vec{y}_0^{(r)}, \Delta^2\vec{y}_1^{(r)}, \ . \ . \ . \ , \Delta^2\vec{y}_{N-1}^{(r)}\right)$$

with

$$\Delta\vec{y}_j^{(r)} = \vec{y}_{j+1}^{(r)} - \vec{y}_j^{(r)}, \qquad \Delta^2\vec{y}_j^{(r)} = \vec{y}_{j+2}^{(r)} - 2\vec{y}_{j+1}^{(r)} + \vec{y}_j^{(r)}$$

The first member $\vec{y}_0$ of the sequence (55) is an initial approximation of $\vec{u}$. The other members are evaluated in the order indicated by the vector equations

$$\vec{y}_{j+1}^{(r)} = \vec{f}\left(y_j^{(r)}\right) \qquad (j = 0,1, \ldots ,N) \tag{56}$$

$$\vec{y}_0^{(r+1)} = \begin{cases} \vec{y}_N^{(r)} - \Delta Y_0 \left(\Delta^2 Y_0\right)^{-1} \Delta\vec{y}_N^{(r)} , & \text{if } \det\left(\Delta^2 Y_0\right) \neq 0 \\ \vec{y}_{N+1}^{(r)} , & \text{if } \det\left(\Delta^2 Y_0\right) = 0 \end{cases} \tag{57}$$

In particular, when $N = 1$, equations (56) and (57) reduce to (52) and (53), respectively.

Steffensen's iteration procedure for the vector case has not been investigated fully from the theoretical point of view. However, in practice, the sub-sequence $\left\{\vec{y}_0^{(k)}\right\}$, for a large number of cases has been found, as in the scalar case, to be quadratically convergent (see ref. 9). Consequently, unless the regular sequence generated by equation (56) is already quadratically convergent, it would be less rapidly convergent than that of Steffensen's. Even in the case when the former is divergent, the latter has been found to be convergent in many cases.

The remainder of this section discusses the utilization of Steffensen's sequence in the acceleration of convergence for algorithm I. The approach here is equally applicable to algorithm II.

Recall that we have from algorithm I the sequences $\left\{\phi_{i,j}'(t_k)\right\}_{j=0}^{\infty}$ $(i = 1,2, \ldots ,n; k = 0,1, \ldots ,M+2)$, where each iterate of each sequence is generated at step 4 of the same algorithm. These sequences can be put in the form of a single sequence of rectangular arrays

$$\left[\psi_{i,k,j}\right] = \begin{bmatrix} \psi_{1,1,j} & \psi_{1,2,j} & \cdots & \psi_{1,M+2,j} \\ \psi_{2,1,j} & \psi_{2,2,j} & \cdots & \psi_{2,M+2,j} \\ \vdots & & & \vdots \\ \psi_{n,1,j} & \psi_{n,2,j} & \cdots & \psi_{n,M+2,j} \end{bmatrix} \tag{58}$$

$(j = 0,1,2, \ldots)$

where

$$\psi_{i,k,j} = \phi'_{i,j}(t_{k-1})$$

The (M + 2) column, being $\phi'_{i,j}(t)$ evaluated at $t = -1$, remains unchanged for all j. Each entry of the initial array $\left[\psi_{i,k,1}\right]$ is the value of the initial approximation to $\phi'_i(t)$ evaluated at t_{k-1}. Specifically, according to the way algorithm I was constructed, this array has the elements

$$\psi_{i,k,1} = \psi_i(t_{k-1},\eta_1,\eta_2, \ . \ . \ . \ ,\eta_n) \qquad (i = 1,2, \ . \ . \ . \ ,n; \ k = 1,2, \ . \ . \ . \ ,M+2)$$

associated with the initial approximation

$$\phi_{i,0}(t) \equiv \eta_i \qquad (i = 1,2, \ . \ . \ . \ ,n)$$

However, from the point of view of convergence, it may be advantageous to interrupt periodically the computation of the iterative procedure and restart it by supplying to the algorithm a new initiating array $[\psi_{i,k,1}]$. The construction of Steffensen's sequence provides a clue as to how this process of iteration modification can be applied effectively. An approach to the solution of this problem might be attempted by first forming various sets of sequences corresponding to different groupings into disjoint subsets of all the entries of the array $[\psi_{i,j,k}]$. In particular, for ease of implementation we consider the four sets of sequences consisting of

(a) The corresponding elements

$$\left\{\psi_{i,k,j}\right\}_{j=0}^{\infty} \qquad (j = 1,2,, \ . \ . \ . \ ,n; \ k = 1,2, \ . \ . \ . \ ,M+2) \tag{59}$$

(b) The total array $\left[\psi_{i,j,k}\right]$

(c) The corresponding rows

$$\left\{\left(\psi_{i,1,j},\psi_{i,2,j}, \ . \ . \ . \ ,\psi_{i,M+2,j}\right)\right\}_{j=0}^{\infty} \qquad (i = 1,2, \ . \ . \ . \ ,n) \tag{60}$$

(d) The corresponding columns

$$\left\{\left(\psi_{1,k,j},\psi_{2,k,j}, \ . \ . \ . \ ,\psi_{n,k,j}\right)^T\right\}_{j=0}^{\infty} \qquad (k = 1,2, \ . \ . \ . \ ,M+2) \tag{61}$$

If certain assumptions, which may or may not be true, are made about each of these sequences, the initial array $\left[\psi_{i,k,1}\right]$ can be modified by any one of four different ways.

Modification of individual entries.- If each sequence of equation (59) can be assumed to be iterates of a single-point scalar iteration function, each element of the initiating array $[\psi_{i,k,1}]$ can be modified by Aitken's formula (53) one at a time. Thus,

$$\left(\psi_{i,k,1}\right)_{new} = \psi_{i,k,3} - \frac{\left(\psi_{i,k,3} - \psi_{i,k,2}\right)^2}{\psi_{i,k,3} - 2\psi_{i,k,2} + \psi_{i,k,1}}$$

For cases $n \geq 2$, however, numerical results show that such a procedure not only failed to hasten convergence but caused the algorithm to diverge. This is not surprising because too much information is lost due to our disregarding implicitly the fact that each element of an array of $\left\{\left[\psi_{i,j,k}\right]\right\}_{j=0}^{\infty}$ is a function of all the elements of the preceding array. The case with $n = 1$ is a special case of modification of columns to be discussed in a later paragraph.

Modification of the total array.- In contrast to the method of modification of the last paragraph, the entire rectangular array $[\psi_{i,j,k}]$ can be considered as an iterate of a single-point vector iteration function of $n(M + 1)$ components. Then the initiating array $[\psi_{i,j,k}]$ can be modified in toto by means of the matrix equation (57). This fully takes into account the dependence property mentioned in the last paragraph. Nevertheless, such a scheme could hardly be considered feasible from the standpoint of efficient computation for the following reasons. Even if n and M are moderate in size, the matrix to be inverted is of a high order $n(M + 1)$. Furthermore, the first modification cannot be effected until after $n(M + 1) + 1$ iterations by which time the algorithm has already converged or will have converged to a solution after a few more iterations. The case for $n = 1$ is more tenable but it simplifies to the method of modification of rows to be discussed presently. (The abbreviations MR and MC in the sequel denote the methods of modification of rows and modification of columns, respectively.)

In view of what is said in the preceding paragraph, any acceleration of convergence procedure involving either modification of individual entries or the total array must be ruled out. On the other hand, the methods MR and MC intermediate in complexity between the two previous ones mentioned are found to be effective in the acceleration of convergence of algorithm I as well as algorithm II of the following section.

Modification of rows (MR).- Suppose each of the sequences (60) is an iterate of a single-point vector iteration function of $M + 1$ components. Then the initiating array $[\psi_{i,k,1}]$ can be modified one row at a time as follows:

Let $[\psi_{i,k,j}]$ $(j = 2,3, \ldots ,M+3)$ be the $M + 2$ rectangular arrays generated by the algorithm for an initiating array $[\psi_{i,k,1}]$. For each i, set

$$\vec{y}_{j-1}^{(r)} = (\psi_{i,1,j}, \psi_{i,2,j}, \ldots , \psi_{i,M+1,j})^T \qquad (j = 1,2, \ldots ,M+3)$$

Compute $\vec{y}_0^{(r+1)}$ by means of the matrix equation (57). The ith row of the new initiating array is then formed by setting

$$\psi_{i,k,1} = y_{k,0}^{(r+1)} \qquad (k = 1,2, \ . \ . \ . \ ,M+1)$$

After all the rows of $[\psi_{i,k,1}]$ have been modified, the iterative process is restarted by returning to step 5 of algorithm I. The testing at step 8 should be bypassed until $[\psi_{i,k,2}]$ has been generated in the following iteration.

Numerical results suggest that unless the sequence of rectangular arrays are already quadratically convergent within M + 2 iterations with no modification, MR modification, in many instances, substantially reduces the number of iterations required for convergence. It is rare when modification is needed more than once.

Since the methods MR and total array modification coincide for n = 1, the latter method can be considered as a special case of the former in this instance.

The feasibility of MR as a convergence acceleration procedure lies in the fact that it does not have the complicating features of total array modification, and unlike entry-wise modification it evidently does make sufficient use of the history of previous iterations to be effective in the acceleration of convergence.

The following method provides another and possibly more effective procedure in the acceleration of convergence for both algorithms I and II.

Modification of columns (MC).- Since MC modification is also intermediate in complexity between the first two methods discussed, it possesses the desirable features of MR modification. Moreover, when n is smaller than M, as often is the case in practice, it may be preferable to modify the initiating array $[\psi_{i,k,1}]$ by the former rather than the latter method. The validity of the statement will become evident in the ensuing discussion.

The assumption made here is that each of the M + 1 sequences (61) are generated by a single-point vector iteration function of n components. Modification in this case is carried out after every n + 1 iterations by making use of the n + 2 rectangular arrays $[\psi_{i,j,k}]$ (j = 1,2, . . . ,n+2) saved in storage. Each column (specifically the kth) of the initiating array is re-evaluated by setting

$$\vec{y}_{j-1}^{(r)} = \left(\psi_{1,k,j}, \psi_{2,k,j}, \ . \ . \ . \ , \psi_{n,k,j}\right)^T \qquad (j = 1,2, \ . \ . \ . \ ,n+2)$$

and determining $\vec{y}_0^{(r+1)}$ by means of the matrix equation (57). The column in question is then replaced with new entries by setting

$$\psi_{i,k,1} = y_{i,0}^{(r+1)} \qquad (i = 1,2, \ldots ,n)$$

The iterative process, after every column of $[\psi_{i,k,1}]$ has been thus modified, as in the case of MR modification, is restarted at step 5 of algorithm I.

The remarks made for MR modification about the testing at step 8 is also applicable here.

Since M is usually set larger than n in practice, to achieve a desired accuracy, the MC method, besides saving storage allows modification to take place sooner and more frequently than modification by the MR method. When this occurs, numerical results show that convergence is accelerated more strongly than by MR modification.

A comparison of the rates of convergence (in terms of number of iterations and machine time applied to the solution of example I) for iteration with MR and MC modification as well as no modification is given in table IV. It shows that both MR and MC methods substantially reduced the number of iterations and machine time required for convergence. It also shows that both the total number of iterations and the total machine time over all of the indicated intervals for MC modification is significantly less than that of MR. This should not, however, mislead us into precluding MR modification as a tool in the acceleration of convergence. Although numerical results show that whenever both methods work, MC modification works better; it is quite conceivable that acceleration of convergence in certain cases may fail for MC modification and work well for MR modification. For specific examples a proper combination of methods (with or without modification) may optimize the computation time required.

TABLE IV.- A COMPARISON OF CONVERGENCE FOR ITERATION WITH AND WITHOUT MODIFICATION (ALGORITHM I).

Modification \ Interval	$3 \le x \le 4$	$4 \le x \le 5$	$5 \le x \le 6$	Total
None	38(5.16)	44(6.00)	53(7.32)	135(18.48)
MC	19(3.42)	18(3.30)	14(2.52)	51(9.24)
MR	25(4.50)	15(3.06)	15(3.12)	55(10.68)

Each entry of the table indicates the number of iterations required by algorithm I to solve example I for the indicated method of modification and interval of integration. The machine time in seconds on the IBM 7094 for the corresponding iteration is given in parentheses. The degree of approximation is $M = 11$ and the prescribed convergence error is $\varepsilon = 0.5\times10^{-10}$.

An Alternate Procedure

Instead of using equations (42) and (43), it is sometimes preferable from the point of view of faster convergence to construct an algorithm to solve a system of n first-order differential equations based on the equations

$$\left.\begin{aligned}
\phi_{1,j+1}(t) &= \eta_1 + \int_{-1}^{t} \psi_1(u,\phi_{1,j},\phi_{2,j}, \ldots ,\phi_{n,j})du \\
\phi_{2,j+1}(t) &= \eta_2 + \int_{-1}^{t} \psi_2(u,\phi_{1,j+1},\phi_{2,j}, \ldots ,\phi_{n,j})du \\
&\vdots \\
\phi_{n,j+1}(t) &= \eta_n + \int_{-1}^{t} \psi_n(u,\phi_{1,j+1},\phi_{2,j+1}, \ldots ,\phi_{n-1,j+1},\phi_{n,j})du
\end{aligned}\right\} \tag{62}$$

with

$$\phi_{i,0}(t) \equiv \eta_i \qquad (i = 1,2, \ldots ,n) \tag{43}$$

These equations differ from equations (42) and (43) in that the most recent information is used at each step. It will be shown in the next section how the specialization of (62) serves as the basis for a more efficient algorithm to solve an nth-order differential equation.

INTEGRATION OF AN nth-ORDER DIFFERENTIAL EQUATION

An nth-order differential equation can be expressed as a system of n first-order differential equations. Consequently, numerical methods for solving a system of first-order equations can be employed to solve an nth-order differential equation (see, e.g., example I). However, in so doing we may be performing many more operations which are otherwise unnecessary if the solution of nth-order differential equation were found without changing it to a system. This is true at least for the case of the Chebyshev series and will be explained in the ensuing discussion.

Let the nth-order differential equation to be solved assume the form

$$\phi^{(n)}(t) = \psi\left(t,\phi,\phi', \ldots ,\phi^{(n-1)}\right), \qquad -1 \leq t \leq 1 \tag{63}$$

with the initial conditions

$$\phi^{(i)}(-1) = \eta_i \qquad (i = 0,1, \ldots ,n-1) \tag{64}$$

To provide the basis for the construction of the algorithm for the solution of an nth-order differential equation, define

$$\phi_1(t) = \phi^{(n-1)}(t)$$

$$\phi_2(t) = \phi^{(n-2)}(t)$$

$$\cdots\cdots\cdots$$

$$\phi_n(t) = \phi(t)$$

Then equations (62) and (43) specialize at once to the equations

$$\left.\begin{aligned}
\phi_{j+1}^{(n-1)}(t) &= \eta_{n-1} + \int_{-1}^{t} \psi\left(u,\phi_j,\phi_j',\ldots,\phi_j^{(n-1)}\right)du \\
\phi_{j+1}^{(n-2)}(t) &= \eta_{n-2} + \int_{-1}^{t} \phi_{j+1}^{(n-1)}(u)\,du \\
&\;\;\vdots \\
\phi_{j+1}(t) &= \eta_0 + \int_{-1}^{t} \phi_{j+1}'(u)\,du
\end{aligned}\right\} \tag{65}$$

with

$$\phi_0^{(i)} \equiv \eta_i \qquad (i = 0,1,\ldots,n-1) \tag{66}$$

The sequence of functions $\left\{\phi_j(t)\right\}_{j=0}^{\infty}$ (as its counterpart $\left\{\phi_{i,j}(t)\right\}_{j=0}^{\infty}$ of eqs. (42) and (43)) is expected to converge uniformly to a function $\phi(t)$ satisfying equations (63) and (64) on the closed interval [-1,1]. Note that unlike the case where an nth-order differential equation is treated as a system of n first-order differential equations, the integration involved in each of equations (65) except the first is applied to the derivatives of the current iteration. Numerical results show that this gives algorithm II a clear advantage over algorithm I in terms of the number of iterations required to solve an nth-order differential equation for a given convergence error (see example II). Furthermore, if the Chebyshev series of

$$\phi_{j+1}^{(n)}(t) = \psi\left(t,\phi_j,\phi_j',\ldots,\phi_j^{(n-1)}\right)$$

is available, the n integrals of (65) can be obtained readily by making use of the relation (B19) between the Chebyshev coefficients of the function and those of its derivatives. Hence, if an algorithm similar to algorithm I is constructed for the nth-order differential equation, the modified interpolating polynomial is used only to approximate the nth derivative during one iteration. This is in contrast to algorithm I which utilizes $Q_N(t)$ n times in the approximation of first derivatives. Since the computation of the modified interpolating polynomial takes up the bulk of the time per iteration, the difference in time spent solving the same differential equation may be considerable.

Algorithm II (for an nth-order equation)

The construction of the algorithm for an approximate solution of an nth-order differential equation, as in the case of algorithm I, is based on the assumption that the sequences of functions $\left\{\phi_j^{(i)}(t)\right\}_{j=0}^{\infty}$ $(i = 0,1, \ldots ,n)$ can be represented accurately by polynomials of sufficiently high degree. Let

$$\sum_{k=0}^{M+n-i}{}' b_k^{(i)} T_k(t) \quad \text{and} \quad \sum_{k=0}^{M+n-i}{}' B_k^{(i)} T_k(t) \tag{67}$$

denote, respectively,

$$\phi_j^{(i)}(t) \quad \text{and} \quad \phi_{j+1}^{(i)}(t) \qquad (i = 0,1, \ldots ,n) \tag{68}$$

For notational simplicity, equality between the respective expressions (67) and (68) is assumed to hold. The approximate solution of equation (63) with initial conditions (64) for a prescribed convergence error ε is obtained as follows:

1. Set

$$t_k = \cos \frac{k\pi}{M + 1}, \qquad k = 0, 1, \ldots , M + 1$$

(This computes the $M + 2$ points where $T_{M+1}(t)$ has an extremum.)

2. Set

$$b_0^{(i)} = 2\eta_i \qquad (i = 0,1, \ldots ,n-1)$$

$$b_k^{(i)} = 0 \qquad (i = 0,1, \ldots ,n-1;\ k = 1,2, \ldots ,M+n-i)$$

(This is equivalent to the initial approximations $\phi^{(i)}(t) \equiv \eta_i$ $(i = 0,1, \ldots , n-1)$.)

3. Compute

$$\phi_j^{(i)}(t_k) = \sum_{r=0}^{M+n-i}{}' b_r^{(i)} T_r(t_k) \qquad (i = 0,1, \ldots, n-1;\ k = 0,1, \ldots, M+1)$$

4. Compute

$$\phi_{j+1}^{(n)}(t_k) = \psi\left[t_k, \phi_j^{(0)}(t_k), \phi_j^{(1)}(t_k), \ldots, \phi_j^{(n-1)}(t_k)\right]$$

$$(k = 0,1, \ldots, M+1)$$

5. Compute (using eq. (B27))

$$B_k^{(n)} = \frac{2}{M+1} \sum_{r=0}^{M+1}{}'' \phi_{j+1}^{(n)}(t_r) T_r(t_k) \qquad (k = 0,1, \ldots, M)$$

(We have here approximated $\phi_{j+1}^{(n)}(t)$ by the Mth degree polynomial

$$\phi_{j+1}^{(n)}(t) = \sum_{k=0}^{M}{}' B_k^{(n)} T_k(t)$$

where according to theorem B6

$$B_k^{(n)} \to \frac{2}{\pi} \int_{-1}^{1} \psi\left[t, \phi_j^{(0)}, \phi_j^{(1)}, \ldots, \phi^{(n-1)}(t)\right] T_k(t)(1 - t^2)^{-1/2}\, dt$$

for a sufficiently large M.)

6. Compute (by means of theorem B4 in appendix B and eq. (A31)) for $i = n-1, n-2, \ldots, 0$

$$B_{M+n-i}^{(i)} = \frac{B_{M+n-i-1}^{(i+1)}}{2(M+n-i)}$$

$$B_k^{(i)} = \frac{B_{k-1}^{(i+1)} - B_{k+1}^{(i+1)}}{2k} \qquad (k = 1,2, \ldots, M+n-i-1)$$

$$B_0^{(i)} = 2\left[\eta_i - \sum_{k=1}^{M+n} (-1)^k B_k^{(i)}\right]$$

(We have in this step obtained the n integrals

$$\sum_{k=0}^{M+n-i}{}' B_k^{(i)} T_k(t) \quad \text{of} \quad \sum_{k=0}^{M}{}' B_k^{(n)} T_k(t) \qquad (i = 0,1, \ldots ,n-1))$$

7. If $\left|B_k^{(i)} - b_k^{(n)}\right| < \varepsilon$ for all k, we are through.

8. Otherwise, set

$$b_k^{(i)} = B_k^{(i)} \qquad (i = 0,1, \ldots ,n;\ k = 0,1, \ldots ,M+n-i)$$

and return to step 3.

Note: To take into consideration the case when $\phi_0^{(n)}(t) \equiv \phi_1^{(n)}(t) \equiv 0$ step 7 should be skipped until the second iteration.

Example II

To provide a basis for comparison for both algorithms, we shall solve the differential equation (48) by algorithm II for the same prescribed convergence error and subintervals as in example I (i.e., $\varepsilon = 0.5\times10^{-10}$ and $x_0 = 0$, $x_1 = 1$, $\ldots$, $x_6 = 6$).

For each subinterval $[x_{r-1},x_r]$ $(r = 0,1, \ldots ,6)$, let $x = ct + d$, $c = \frac{1}{2}(x_{r+1} - x_r)$, $d = \frac{1}{2}(x_r + x_{r+1})$. Thus,

$$F^{(i)}(x) = F^{(i)}(ct + d) = 2^i\phi^{(i)}(t) \qquad (i = 0,1,2,3)$$

Substitution into (48) then yields

$$\phi'''(t) = \frac{1}{2}\left\{[\phi'(t)]^2 - 2\phi(t)\phi''(t) - \frac{1}{4}\right\}$$

the form required by the algorithm. The solution is found as six initial-value problems corresponding to the six subintervals. The initial conditions used for each subinterval are determined by the solution at the end point of the previous subinterval except the first where

$$\phi^{(i)}(-1) = \frac{1}{2^i} F^{(i)}(0) \qquad (i = 0,1, \ldots ,n-1)$$

A comparison of computer results obtained by algorithm II with that of algorithm I shows agreement to 10 decimal places. Computer results also show that the coefficients $B_k^{(i)}$ $(i = 0,1, \ldots ,n)$ exhibit no change to 10 decimal places for $M \geq 11$. Consequently, from the discussion in the preliminary analysis section the approximate Chebyshev series satisfies the differential equation with an error bound of order $\varepsilon = 0.5\times10^{-10}$.

A comparison of tables III and V reveals that the number of iterations required for the solution of the aforementioned differential equation is consistently less for algorithm II than for algorithm I.

TABLE V.- THE INFLUENCE OF M ON THE CONVERGENCE OF ALGORITHM II

M \ Interval	$0 \leq x \leq 1$	$1 \leq x \leq 2$	$2 \leq x \leq 3$	$3 \leq x \leq 4$	$4 \leq x \leq 5$	$5 \leq x \leq 6$
11	10	17	24	33	39	40
12	10	17	23	31	37	37
13	10	17	23	28	34	27
14	10	17	23	26	27	26
15	10	17	23	27	28	26

Each entry of this table indicates the number of iterations algorithm II required to solve equation (48) having the indicated M and interval of integration. The convergence criterion is $\varepsilon = 0.5\times10^{-10}$. Since the approximate Chebyshev coefficients agree to 10 decimal places for all $M \geq 11$, the accuracy of the solution for each M is good to 10 decimal places.

Remarks on Convergence

Since a polynomial is used to approximate

$$\phi_{j+1}^{(n)}(t) = \psi\left(t,\phi_j,\phi_j', \ldots ,\phi_j^{(n-1)}\right)$$

the convergence of algorithm II, as in the case of algorithm I, may be influenced by the degree of the approximating polynomial M. Examination of table V shows that while the number of iterations needed for convergence for the first three intervals remains essentially unchanged, the number of iterations in the remaining three intervals required for convergence is essentially a decreasing function of M. Thus by a judicious choice of M the computing time spent for a particular problem may be minimized. However, when such a choice is unavailable, it is probably better to pick a higher M than is thought necessary so that convergence can be achieved within a tolerable number of iterations. For example, in the case of example II, it would have been better to pick $M = 15$ instead of 11.

It will be shown next that the convergence of algorithm II for a fixed M can be accelerated by methods already proposed for a system of n first-order differential equations.

Acceleration of Convergence

Note first that the sequences $\left\{\phi_j^{(n)}(t_k)\right\}_{j=1}^{\infty}$ $(k = 0,1, \ldots ,M + 1)$ can be considered as a single sequence of one-dimensional array

$$\left(\psi_{j,1},\psi_{j,2}, \ldots ,\psi_{j,M+2}\right) \qquad (j = 1,2, \ldots) \tag{69}$$

where

$$\psi_{j,k} = \phi_j^{(n)}(t_{k-1}) \tag{70}$$

Thus if one thinks of (69) as a rectangular array of one row and $M + 2$ columns, the methods of MR and MC modification proposed for algorithm I can also be used in the acceleration of convergence for the nth-order differential equation. The initiating array $\left(\psi_{1,1},\psi_{2,1}, \ldots ,\psi_{M+2,1}\right)$ is re-evaluated for every $M + 2$ or every two iterations depending upon whether MR or MC modification is utilized. Since MR modification is applied to a single row and MC is actually a modification of individual entries here, the number of operations involved compare with that of algorithm I (applied to the solution of the same nth-order differential equation) should be considerably less.

Table VI shows that the number of iterations as well as the machine time required in the solution of example II for both MR and MC methods is considerably less than that of iteration without modification. It also shows that MC modification has a definite edge over MR modification both in the number of iterations and in machine time required for convergence. However, the comments applied to algorithm I about not precluding MR modification as a tool in the acceleration of convergence should also be heeded in the present case.

Table VII is based on the data of tables IV and table VI and illustrates the advantage of algorithm II over algorithm I in the amount of machine time spent on solving the same differential equation. It shows that in every case, whether modification is involved or not, the machine time required by algorithm I is at least 1-1/2 times that of algorithm II. In the case of MC modification the ratio of the machine time of algorithm I to that of algorithm II is as large as 2.86.

TABLE VI.- A COMPARISON OF THE CONVERGENCE FOR ITERATION WITH AND WITHOUT MODIFICATION (ALGORITHM II)

Modification \ Interval	$3 \le x \le 4$	$4 \le x \le 5$	$5 \le x \le 6$	Total
None	33(3.25)	39(3.86)	40(3.98)	112(11.09)
MC	16(1.80)	14(1.57)	8(0.88)	38(4.25)
MR	18(2.21)	15(1.92)	15(1.93)	48(6.05)

Each entry of this table indicates the number of iterations required when using algorithm II to solve example II for the indicated method of modification and interval of integration. The machine time in seconds on the IBM 7094 for the corresponding iteration is given in parentheses. The degree of approximation is $M = 11$ and the convergence criterion is $\varepsilon = 0.5\times10^{-10}$.

TABLE VII.- RATIOS OF MACHINE TIME OF ALGORITHM I TO ALGORITHM II

Modification \ Interval	$3 \le x \le 4$	$4 \le x \le 5$	$5 \le x \le 6$
None	1.59	1.55	1.84
MC	1.87	2.10	2.86
MR	2.03	1.59	1.62

Each entry of this table gives the ratio of the machine time of algorithm I to that of algorithm II based on tables IV and VI.

Ames Research Center
National Aeronautics and Space Administration
Moffett Field, Calif., 94035, May 15, 1969

APPENDIX A

CHEBYSHEV POLYNOMIALS

This appendix is included to facilitate the discussion of the approximation of functions in terms of Chebyshev polynomials. It also provides the minimum of the necessary working tools for most numerical work involving these polynomials. Most of the properties can be found in the works of Rivlin (ref. 10) and Lanczos (ref. 11). Clenshaw's summation formula for Chebyshev series is generalized here for the summation of sequences obeying an nth-order linear recurrence relation (see theorem A1).

Some Properties of Chebyshev Polynomials

Definition.- The polynomial of degree k defined by

$$T_k(t) = \cos(k \text{ arc cos } t) , \qquad -1 \le t \le 1 \tag{A1}$$

is called *the Chebyshev polynomial of the first kind of order* k or simply a Chebyshev polynomial.

As an immediate consequence of this definition we have

Property A1. If $k \ge 2$, then

$$T_k(t) = 2tT_{k-1}(t) - T_{k-2}(t) \tag{A2}$$

with

$$T_0(t) = 1, \qquad T_1(t) = t$$

From the recurrence relation (A2) we can generate Chebyshev polynomials up to any order n by setting $k = 2, 3, \ldots, n$.

By the change of variable $t = \cos\theta$ one can demonstrate that the sequence of Chebyshev polynomials $\left\{T_k(t)\right\}_{k=0}^{\infty}$ is orthogonal on $[-1,1]$ with respect to the weight function $w(t) = (1-t^2)^{-1/2}$, that is,

Property A2. For any two Chebyshev polynomials $T_m(t)$ and $T_k(t)$, the following conditions hold.

$$\int_{-1}^{1} T_m(t)T_k(t)(1 - t^2)^{-1/2}\, dt = \begin{cases} 0 & \text{if } m \ne k \\ \frac{\pi}{2} & \text{if } m = k \ne 0 \\ \pi & \text{if } m = k = 0 \end{cases} \tag{A3}$$

Consequently, any results deduced for general classes of orthogonal polynomials also hold for Chebyshev polynomials.

<u>Property A3 (Orthogonality with Respect to Summation).</u>

Let

$$t_j = \cos \frac{j\pi}{N + 1} \qquad (j = 0,1, \ldots ,N+1) \tag{A4}$$

If $T_m(t)$ and $T_k(t)$ are any pair of Chebyshev polynomials of orders zero through $N + 1$, then

$$\sum_{j=0}^{N+1}{}'' T_m(t_j)T_k(t_j) = \begin{cases} N + 1 & \text{if } m = k = 0 \text{ or } N + 1 \\ (N + 1)/2 & \text{if } m = k \neq 0 \text{ or } N + 1 \\ 0 & \text{if } m \neq k \end{cases} \tag{A5}$$

Furthermore, if $T_r(t)$ is a Chebyshev polynomial of any order

$$\sum_{j=0}^{N+1}{}'' T_r(t_j) = \begin{cases} N + 1 & \text{if } r = 2s(N + 1) , \quad s = 0,1,2, \ldots \\ 0 & \text{if } r \neq 2s(N + 1) , \quad s = 0,1,2, \ldots \end{cases} \tag{A6}$$

<u>Property A4.</u> For all $k \geq 1$, $T_k(t)$ has zeros at

$$t = \cos \frac{(2j + 1)\pi}{2k} \qquad (j = 0,1, \ldots ,k-1) \tag{A7}$$

and extrema $(-1)^j$ at

$$t = \frac{j\pi}{k} \qquad (j = 0,1, \ldots ,k) \tag{A8}$$

Property A3 together with equation (A8) is instrumental to the discussion of the "Modified Interpolation" of appendix B. The following property is useful in the integration of a function represented by a series of Chebyshev polynomials:

<u>Property A5.</u>

$$\int T_k(t)dt = \begin{cases} T_1(t) & , \quad k = 0 \\ T_2(t)/4 & , \quad k = 1 \\ 1/2\left(\dfrac{T_{k+1}(t)}{k + 1} - \dfrac{T_{k-1}(t)}{k - 1}\right) & , \quad k \geq 2 \end{cases} \tag{A9}$$

Summation of Series of Chebyshev Polynomials

We consider next the summation of a finite series of Chebyshev polynomials of the form

$$S_N(t) = \sum_{k=0}^{N}{}' B_k T_k(t) \tag{A10}$$

where B_k are constants. The evaluation of $S_N(t)$ can be efficiently performed by particularization of the following theorem (see also ref. 12):

<u>Theorem A1</u>. If

$$S = \sum_{k=0}^{N} a_k u_k \tag{A11}$$

where u_k obeys the nth-order recurrence relation

$$u_k = \sum_{j=1}^{n} r_{j,k} u_{k-j} , \qquad k \geq n \tag{A12}$$

then S can be evaluated by the formula

$$S = c_0 u_0 + \sum_{k=1}^{n-1} c_k \left(u_k - \sum_{j=1}^{k} r_{j,k} u_{k-j} \right) \tag{A13}$$

where $c_0, c_1, \ldots, c_{n-1}$ are determined from the recurrence formula

$$\left.\begin{aligned} & c_{N+1} = c_{N+2} = \ldots = c_{N+n} = 0 \\ & c_k = a_k + \sum_{j=1}^{n} r_{j,k+j} c_{k+j} \qquad (k = N, N-1, \ldots, 0) \end{aligned}\right\} \tag{A14}$$

The theorem can be established as follows: Replace a_k in equation (A11) by equation (A14) so that

$$S = \sum_{k=0}^{N} \left(c_k - \sum_{j=1}^{n} r_{j,k+j} c_{k+j} \right) u_k$$

or

$$S = \sum_{k=0}^{N} c_k u_k - \sum_{j=1}^{n} \sum_{k=0}^{N} r_{j,k+j} c_{k+j} u_k$$

Setting now $k + j = m$ in the double sum and noting that $c_k = 0$ for $k = N + 1, N + 2, \ldots, N + n$, we have

$$S = \sum_{k=0}^{N} c_k u_k - \sum_{j=1}^{n} \sum_{m=j}^{N} r_{j,m} c_m u_{m-j}$$

or by inverting the order of summation for the double sum, one obtains

$$S = \sum_{k=0}^{N} c_k u_k - \sum_{m=1}^{n-1} \sum_{j=1}^{m} r_{j,m} c_m u_{m-j} + \sum_{m=n}^{N} \sum_{j=1}^{n} r_{j,m} c_m u_{m-j}$$

The collecting of terms in c_m then yields

$$S = c_0 u_0 + \sum_{m=1}^{n-1} \left(u_m - \sum_{j=1}^{m} r_{j,m} u_{m-j} \right) c_m + \sum_{m=n}^{N} \left(u_m - \sum_{j=1}^{n} r_{j,m} u_{m-j} \right) c_m \qquad \text{(A15)}$$

But by equation (A12)

$$u_m - \sum_{j=1}^{n} r_{j,m} u_{m-j} = 0 \qquad \text{for} \quad m \geq n$$

This proves that equation (A13) holds.

Useful special cases:

1. For $n = 1$, we have

$$u_k = r_k u_{k-1} \qquad (k \geq 1) \qquad \text{(A16)}$$

so S of (A13) becomes

$$S = c_0 u_0 \qquad \text{(A17)}$$

with

$$c_{N+1} = 0 , \qquad c_k = a_k + r_{k+1} c_{k+1} \qquad (k = N, N-1, \ldots, 0) \qquad \text{(A18)}$$

2. For $n = 2$, we have

$$u_k = r_{1,k}u_{k-1} + r_{2,k}u_{k-2} \tag{A19}$$

so S of equation (A13) becomes

$$S = c_0u_0 + c_1(u_1 - r_{1,1}u_0) \tag{A20}$$

with

$$\left.\begin{aligned} &c_{N+1} = c_{N+2} = 0 \\ &c_k = a_k + r_{1,k+1}c_{k+1} + r_{2,k+2}c_{k+2} \\ &(k = N, N-1, \ldots, 0) \end{aligned}\right\} \tag{A21}$$

The following properties are applications of the special case of $n = 2$ to sums of Chebyshev polynomials:

<u>Property A6</u>. If

$$S(t) = \sum_{k=0}^{N}{}' B_kT_k(t) \tag{A22}$$

then also

$$S(t) = \frac{B_0}{2} + c_1t - c_2 \tag{A23}$$

where c_1 and c_2 are generated by the recurrence relations

$$\left.\begin{aligned} &c_{N+1} = c_{N+2} = 0 \\ &c_k = B_k + 2tc_{k+1} - c_{k+2} \qquad (k = N, N-1, \ldots, 1) \end{aligned}\right\} \tag{A24}$$

(This formula follows from eq. (A21) and the fact that

$$T_k(t) = 2tT_{k-1}(t) - T_{k-2}(t) \qquad (k \geq 2))$$

<u>Property A7</u>. If

$$S(t) = \sum_{k=0}^{N}{}' B_kT_{2k}(t) \tag{A25}$$

then also

$$S(t) = \frac{B_0}{2} + c_1(2t^2 - 1) - c_2 \tag{A26}$$

where c_1 and c_2 are generated by the recurrence relations

$$\left.\begin{aligned} &c_{N+1} = c_{N+2} = 0 \\ &c_k = B_k + 2(2t^2 - 1)c_{k+1} - c_{k+2} \qquad (k = N, N-1, \ldots, 1) \end{aligned}\right\} \tag{A27}$$

(The formula follows from eq. (A21) and the fact that

$$T_{2k}(t) = 2(2t^2 - 1)T_{2k-2} - T_{2k-2}(t) - T_{2k-4}(t) \qquad (k \geq 2))$$

<u>Property A8</u>. If

$$S(t) = \sum_{k=1}^{N} B_k T_{2k+1}(t) \tag{A28}$$

then also

$$S(t) = (2t^2 - 1)(c_1 - c_2) \tag{A29}$$

where c_1 and c_2 are generated by the recurrence relations

$$\left.\begin{aligned} &c_{N+1} = c_{N+2} = 0 \\ &c_k = B_k + 2(2t^2 - 1)c_{k+1} - c_{k+2} \qquad (k = N, N-1, \ldots, 1) \end{aligned}\right\} \tag{A30}$$

(The formula follows from equation (A21) and the fact that

$$T_{2k+1}(t) = 2(2t^2 - 1)T_{2k-1}(t) - T_{2k-3}(t))$$

Formulas for evaluating $S_N(t)$ of equation (A10) at special values of t are given as follows:

<u>Property A9</u>. If

$$S_N(t) = {\sum_{k=0}^{N}}' B_k T_k(t) , \qquad -1 \leq t \leq 1 \tag{A10}$$

then

$$S_N(-1) = \sum_{k=0}^{N}{}' (-1)^k B_k \tag{A31}$$

$$S_N(0) = \sum_{k=0}^{N}{}' (-1)^k B_{2k} \tag{A32}$$

and

$$S_N(1) = \sum_{k=0}^{N}{}' B_k \tag{A33}$$

APPENDIX B

THE CHEBYSHEV SERIES AND APPROXIMATION BY MODIFIED INTERPOLATION

The Best Approximating Polynomial and the Chebyshev Series

Let $f(x)$ be a continuous function defined on a closed interval $[a,b]$ and let ε be a prescribed positive number. The existence of a polynomial $P(x)$ for which

$$\max_{a \leq x \leq b} \left| P(x) - f(x) \right| < \varepsilon \tag{B1}$$

is given by a well-known theorem of Weierstrass. We introduce here the concept of the best approximating polynomial to facilitate later discussion. (For a detailed discussion of the best approximating polynomial and its properties, see ref. 13.)

Definition.- Let D_N denote the set of polynomials of degree $\leq N$. A polynomial $P^*(x) \varepsilon D_N$ having the *maximum residual*:

$$\max_{a \leq x \leq b} \left| f(x) - P^*(x) \right| = \min_{P \varepsilon D_N} \left(\max_{a \leq x \leq b} \left| f(x) - P(x) \right| \right) \tag{B2}$$

is called a *best approximating polynomial* of degree N of $f(x)$ in the Chebyshev sense.

Definition.- The quantity

$$E_N(f) = \min_{P \varepsilon D_N} \left(\max_{a \leq x \leq b} \left| f(x) - P(x) \right| \right) \tag{B3}$$

is called the smallest deviation of the polynomials of D_n from $f(x)$ or the *minimax*.

The best approximating polynomial always exists and is unique. It is completely characterized by the theorem of P. L. Chebyshev.

Theorem B1. Let $f(x)$ be a function continuous on $[a,b]$. Then any polynomial $P(x) \varepsilon D_N$ is the best approximating polynomial if and only if there exist $N + 2$ points

$$a \leq x_1 < x_2 < \ . \ . \ . < x_{N+2} \leq b \tag{B4}$$

for which

$$\left| f(x_i) - P(x_i) \right| = \max_{a \leq x \leq b} \left| f(x) - P(x) \right| = E \qquad (i = 1,2, \ . \ . \ . ,N+2) \tag{B5}$$

with the function $f(x) - P(x)$ alternating in sign at consecutive values of x_i.

A useful consequence of the above theorem is:

Corollary.- If $f(x)$ is continuous on $[a,b]$ and if for any $Q(x)$ in D_N the function $f(x) - Q(x)$ alternates in sign on a set of $N + 2$ distinct points

$$a \leq x_1 < x_2 < \ldots < x_{N+2} \leq b \tag{B6}$$

with

$$|f(x_i) - Q(x_i)| = M \tag{B7}$$

then

$$M \leq E_N(f) \tag{B8}$$

Let $\phi(t)$ be continuous on the closed interval $[-1,1]$. We shall be interested in the expansion

$$\phi(t) = {\sum_{k=0}^{\infty}}' A_k T_k(t) \;, \qquad -1 \leq t \leq 1 \tag{B9}$$

where $T_k(t)$ are Chebyshev polynomials (see appendix A) defined by

$$T_k(t) = \cos(k \text{ arc } \cos t) \tag{B10}$$

Definition.- In the particular case where

$$A_k = \frac{2}{\pi} \int_{-1}^{1} \phi(t) T_k(t) (1 - t^2)^{-1/2} \, dt \tag{B11}$$

the series (B9) is known as the *Chebyshev series* for $\phi(t)$, and the coefficients A_k are called *Chebyshev coefficients*. (See refs. 10 and 13 for a detailed treatment of expansion in Chebyshev series, and of certain relations between the best approximating polynomial and the Chebyshev series.)

The following exhibits a large class of functions which have uniformly convergent Chebyshev series expansions:

Theorem B2.- If $\phi(t)$ satisfies the Hölder condition, that is, if there exists constants M and α such that, for all t_1 and t_2 in $[-1,1]$

$$|\phi(t_1) - \phi(t_2)| \leq M|t_1 - t_2|^{\alpha} \qquad (\alpha > 0) \tag{B12}$$

then $\phi(t)$ can be expanded in a uniformly convergent Chebyshev series. (For efficient computation, however, the given function should have stronger properties such as differentiability. Consequently, prior to expansion in Chebyshev series, it may be necessary to provide such properties by suitable transformations or subdivision of the interval of definition.)

Let the partial sum formed by the first $N + 1$ terms of the Chebyshev series of a function $\phi(t)$ be denoted by

$$S_N(t) = \sum_{k=0}^{N}{}' A_k T_k(t) \tag{B13}$$

and the maximum error of $S_N(t)$ by

$$\sigma_N(\phi) = \max_{-1\le t\le 1} |\phi(t) - S_N(t)| \tag{B14}$$

we note here that, since $S_N(t)$ is a linear combination of polynomials at most degree N, the partial sum $S_N(t) \varepsilon D_N$.

The following theorem gives an important inequality between the minimax error $E_N(\phi)$ and the error $\sigma_N(\phi)$:

<u>Theorem B3 (A. Lesbesque)</u>.- If $\phi(t)$ is continuous on $[-1,1]$, then

$$\sigma_N(\phi) \le (3 + \log N) E_N(\phi) \tag{B15}$$

The above inequality means that for practical purposes the truncated Chebyshev series is just as good as the best approximating polynomial.

Some useful inequalities for a function $\phi(t)$ which is expandable in a uniformly convergent Chebyshev series are:

$$E_N(\phi) \le \sigma_N(\phi) \le \sum_{k=N+1}^{\infty} |A_k| \tag{B16}$$

$$\max_{-1\le t\le 1} |P^*(t) - S_N(t)| \le 2 \sum_{k=N+1}^{\infty} |A_k| \tag{B17}$$

$$\sqrt{1/2 \sum_{k=N+1}^{\infty} A_k^2} \le E_N(\phi) \le \sum_{k=N+1}^{\infty} |A_k| \tag{B18}$$

A useful relation between the Chebyshev coefficients of a function and the Chebyshev coefficients of its derivative is given by:

Theorem B4.- Let $\phi(t)$ be defined on $[-1,1]$. If $\phi'(t)$ is integrable, then

$$A_{k-1}^{(1)} - A_{k+1}^{(1)} = 2kA_k^{(0)} \qquad (k = 1,2, \ldots) \tag{B19}$$

where $A_k^{(0)}$ and $A_k^{(1)}$ are Chebyshev coefficients of $\phi(t)$ and $\phi'(t)$, respectively.

(The validity of eq. (B19) can be demonstrated by making the change of variable $t = \cos\theta$ in eq. (B11) and then integrating by parts.)

Approximation by Modified Interpolation

Let $f(x)$ be continuous on $[a,b]$ and let D_N denote the set of all polynomials of at most, degree N. Recall that the best approximating polynomial of $f(x)$ as defined in the preceding section is the polynomial $P^*(x) \epsilon D_N$ for which

$$\max_{a \le x \le b} |f(x) - P^*(x)| = \min_{P \epsilon D_N} \left(\max_{a \le x \le b} |f(x) - P(x)| \right)$$

It has also been stated that the necessary and sufficient condition for a polynomial $P(x) \epsilon D_N$ to be a best approximating polynomial is that there exist in $[a,b]$ at least $N + 2$ points

$$x_1 < x_2 < \ldots < x_{N+2}$$

for which

$$|f(x_i) - P(x_i)| = \max_{a \le x \le b} |f(x) - P(x_i)| = E \qquad (i = 1,2, \ldots ,N+2)$$

with $f(x) - P(x)$ alternating in sign at consecutive values of x_i. In view of this and the fact that $T_{N+1}(t)$ assumes its extrema $(-1)^j$ at the points

$$t_j = \cos \frac{j\pi}{N + 1}, \qquad j = 0,1, \ldots ,N+1$$

it is easy to see that the following holds:

Lemma.- If

$$G_{N+1}(t) = \sum_{k=0}^{N+1}{}' \beta_k T_k(t) \qquad (-1 \le t \le 1) \tag{B20}$$

then the polynomial

$$H_N(t) = \sum_{k=0}^{N}{}' \beta_k T_k(t) \tag{B21}$$

is the best approximating polynomial of at most, degree N to $G_{N+1}(t)$, and the function $\beta_{N+1}T_{N+1}(t)$ assumes its extrema $(-1)^j \beta_{N+1}$ at the points

$$t_j = \cos \frac{j\pi}{N+1} \qquad (j = 0,1,2, \ldots ,N+1) \tag{B22}$$

The interpolating polynomial $P_{N+1}(t)$.- Attention is now turned to the interpolating polynomial used by Clenshaw in reference 3. A modified version of it is a basic tool of the algorithms of this report. His polynomial assumes the form

$$P_{N+1}(t) = \sum_{k=0}^{N+1}{}'' B_k T_k(t) \tag{B23}$$

and interpolates a given function $\phi(t)$ at $N + 2$ points of t, given by equation (B22). Thus the coefficients B_k of $P_{N+1}(t)$ can be determined by solving directly the linear system

$$\sum_{k=0}^{N+1}{}'' B_k T_k(t_j) = \phi(t_j) \qquad (j = 0,1,2, \ldots ,N+1) \tag{B24}$$

of $N + 2$ equations in $N + 2$ unknowns. However, these equations can best be solved by taking advantage of the orthogonality property of Chebyshev polynomials with respect to summation. First multiply each one of the systems of equations (B24) by $T_m(t_j)$ and then divide the first and last equations by 2. Hence, upon adding the resulting equations, we have

$$\sum_{j=0}^{N+1}{}'' \phi(t_j)T_m(t_j) = \sum_{j=0}^{N+1}{}'' \sum_{k=0}^{N+1}{}'' B_k T_k(t_j)T_m(t_j)$$

Changing the order of summation yields

$$\sum_{j=0}^{N+1}{}'' \phi(t_j)T_m(t_j) = \sum_{k=0}^{N+1}{}'' B_k \sum_{j=0}^{N+1}{}'' T_k(t_j)T_m(t_j)$$

and thus, by equation (A5),

$$B_k = \frac{2}{N+1} \sum_{j=0}^{N+1}{}'' \phi(t_j)T_k(t_j) \qquad (k = 0,1, \ldots ,N+1) \tag{B25}$$

Since

$$T_k(t_j) = \cos\frac{kj\pi}{N+1} = T_j(t_k) \tag{B26}$$

the coefficient B_k can be written also as

$$B_k = \frac{2}{N+1} \sum_{j=0}^{N+1}{}'' \phi(t_j)T_j(t_k) \qquad (k = 0,1, \ldots ,N+1) \tag{B27}$$

We note here that equation (B27) is more desirable than (B25) from the point of view of computation. Since B_k is a finite series of Chebyshev polynomials evaluated at t_k, it can be readily calculated by means of Property A6.

Some important properties of the approximating polynomial $P_{n+1}(t)$ are given in the following theorem.

<u>Theorem B5</u>.- If a function $\phi(t)$ is continuous on [-1,1], and $P_{N+1}(t)$ is the approximating polynomial defined by equations (B23) and (B27), then for each k

$$B_k = A_k + \sum_{r=1}^{\infty}\left(A_{2r(N+1)-k} + A_{2r(N+1)+k}\right) \tag{B28}$$

where A_k are the Chebyshev coefficients of $\phi(t)$, and

$$\frac{1}{2}\left|B_{N+1}\right| \le E_N(\phi) \tag{B29}$$

Equation (B28) can be demonstrated to hold by means of equations (A5) and (A6). To prove (B29) note that by the lemma at the beginning of the section

$$Q_N(t) = \sum_{k=0}^{N}{}' B_kT_k(t)$$

is the best approximating polynomial of maximum degree N to $P_{N+1}(t)$. The residual function $P_{N+1}(t) - Q_N(t) = (1/2)B_{N+1}T_{N+1}(t)$ assumes its extrema

$(-1)^j(B_{N+1}/2)$ at the points $t_j = \cos[j\pi/(N+1)]$, $(j = 0,1, \ldots ,N+1)$. Now since $\phi(t)$ coincides with $P_{N+1}(t)$ at the same $N + 2$ points, the residual function $\phi(t) - Q_N(t)$ also assumes the values $(-1)^j(B_{N+1}/2)$ at these points. Recall that if $f(x)$ is continuous on $[a,b]$ and if for any $Q_N(x)$ in D_N the residual function $f(x) - Q_N(x)$ alternates in sign on a set of distinct points $a \leq x_1 < x_2 < \ldots < x_{N+2} \leq b$ with $|f(x_i) - Q_N(x_i)| = M$, then $M \leq E_N(f)$. Hence it follows that $(1/2)|B_{N+1}| \leq E_N(\phi)$. This proves equation (B29) and also the theorem.

The modified interpolating polynomial $Q_N(t)$.- The polynomial $P_{N+1}(t)$ is an interpolating polynomial having the $N + 2$ extrema of $T_{N+1}(t)$ as the points of interpolation. Let $P_N(t)$ denote the interpolating polynomial having $N + 1$ extrema of $T_N(t)$ as the points of interpolation. We discuss next the modified interpolating polynomial $Q_N(t)$ formed by truncating the last term of $P_{N+1}(t)$; that is,

$$Q_N(t) = \sum_{k=0}^{N}{}' B_k T_k(t) \tag{B30}$$

and also point out why $Q_N(t)$ is preferred over $P_N(t)$ as an approximating polynomial.

Consider first the maximum deviation of $P_{N+1}(t)$ from $S_{N+1}(t)$, the first $N + 2$ terms of the Chebyshev series. Writing

$$P_{N+1}(t) - S_{N+1}(t) = \sum_{k=0}^{N+1}{}' (B_k - A_k)T_k(t) + (\tfrac{1}{2} B_{N+1} - A_{N+1})T_{N+1}(t)$$

and taking absolute values one obtains

$$|P_{N+1}(t) - S_{N+1}(t)| \leq \sum_{k=0}^{N}{}' |B_k - A_k| + |\tfrac{1}{2} B_{N+1} - A_{N+1}| \tag{B31}$$

But we have by equation (B28) the inequalities

$$\left.\begin{aligned}
\tfrac{1}{2}|B_0 - A_0| &\le \phantom{|A_{2N+1}| +{}} |A_{2N+2}| \phantom{{}+|A_{4N+3}|} + |A_{4N+4}| + \cdots \\
|B_1 - A_1| &\le |A_{2N+1}| + |A_{2N+3}| + |A_{4N+3}| + |A_{4N+5}| + \cdots \\
&\;\;\vdots \\
|B_{N-1} - A_{N-1}| &\le |A_{N+3}| + |A_{3N+1}| + |A_{3N+5}| + |A_{5N+3}| + \cdots \\
|B_N - A_N| &\le |A_{N+2}| + |A_{3N+2}| \phantom{+{}} |A_{3N+4}| + |A_{5N+4}| + \cdots \\
|\tfrac{1}{2} B_{N+1} - A_{N+1}| &\le \phantom{|A_{N+2}| +{}} |A_{3N+3}| \phantom{{}+|A_{3N+4}|} + |A_{5N+5}| + \cdots
\end{aligned}\right\} \tag{B32}$$

Add the $N + 2$ equations and sum the right member according to ascending indexes to yield

$$\sum_{k=0}^{N}{}' |B_k - A_k| + |\tfrac{1}{2} B_{N+1} - A_{N+1}| \le \sum_{k=N+2}^{\infty} |A_k| \tag{B33}$$

It follows from inequalities (B31) and B(33) that

$$\max_{-1\le t\le 1} |P_{N+1}(t) - S_{N+1}(t)| \le \sum_{k=N+2}^{\infty} |A_k| \tag{B34}$$

The application of the triangle inequality

$$|\phi(t) - P_{N+1}(t)| \le |\phi(t) - S_{N+1}(t)| + |S_{N+1}(t) - P_{N+1}(t)|$$

together with inequalities (B17) and (B33) yields the error bound for $P_{N+1}(t)$

$$\max_{-1\le t\le 1} |\phi(t) - P_{N+1}(t)| \le 2 \sum_{k=N+2}^{\infty} |A_k| \tag{B35}$$

We consider now the modified polynomial $Q_N(t)$ formed by the first $N + 1$ terms of $P_{N+1}(t)$.

Since

$$|Q_N(t) - S_N(t)| \le \sum_{k=0}^{N}{}' |B_k - A_k|$$

it follows easily from (B32) that

$$\max_{-1\le t\le t} |\phi(t) - Q_N(t)| \le \sigma_N(\phi) + \max_{-1\le t\le t} |Q_N(t) - S_N(t)| \le |A_{N+1}| + 2\sum_{k=N+2}^{\infty} |A_k| \tag{B36}$$

On the other hand, we have by replacing $N + 1$ of inequality (B35) by N

$$\max_{-1\le t\le 1} |\phi(t) - P_N(t)| \le \sum_{k=N+1}^{\infty} |A_k| \le 2\,|A_{N+1}| + \sum_{k=N+2}^{\infty} |A_k| \tag{B37}$$

Hence it can be seen from inequalities (B36) and (B37) that the maximum error for $P_N(t)$ for sufficiently large N can be two times larger than that of $Q_N(t)$. Moreover, since

$$|\phi(t) - Q_N(t)| \le |\phi(t) - P_{N+1}(t)| + |P_{N+1}(t) - Q_N(t)|$$

we have from the fact that $|T_{N+1}(t)| \le 1$ and the inequality (B35) that

$$\max_{-1\le t\le 1} |\phi(t) - Q_N(t)| \le \frac{1}{2}\,|B_{N+1}| + 2\sum_{k=N+2}^{\infty} |A_k| \tag{B38}$$

But

$$\frac{1}{2}\,|B_{N+1}| \le E_N(\phi) \le \max_{-1\le t\le 1} |\phi(t) - Q_N(t)|$$

Consequently,

$$\frac{1}{2}\,|B_{N+1}| \le \max_{-1\le t\le 1} |\phi(t) - Q_N(t)| \le \frac{1}{2}\,|B_{N+1}| + 2\sum_{k=N+2}^{\infty} |A_k| \tag{B39}$$

Note that since

$$\phi(t_j) - {\sum_{k=0}^{N}}' B_k T_k(t_j) = \frac{1}{2} B_{N+1} T_{N+1}(t_j)$$

by (24), the residual function $\phi(t) - Q_N(t)$ also assumes the values $(-1)^j (B_{N+1}/2)$ at $N + 2$ points given by (B22).

Thus if

$$2 \sum_{k=N+2}^{\infty} |A_k|$$

is small relative to $(1/2)|B_{N+1}|$ (and this is often the case in practice), we have

$$\max_{-1 \leq t \leq 1} |\phi(t) - Q_N(t)| \approx \frac{1}{2} |B_{N+1}|$$

and $Q_N(t)$ closely approximates the best approximating polynomial of $\phi(t)$.

APPENDIX C

FORTRAN IV SUBROUTINES FOR ALGORITHMS I AND II

Subroutine AL ALG1

Identification

AL ALG1, Chebyshev Series Integration of a System of n First-Order Nonlinear Differential Equations
FORTRAN IV, Double-Precision Subroutine

Purpose

This subroutine is used to generate an approximate Chebyshev series solution for a system of n first-order nonlinear differential equations with n initial conditions. The differential equations are of the form

$$\frac{d\phi_i}{dt} = \psi(t,\phi_1,\phi_2, \ . \ . \ . \ ,\phi_n) \ , \qquad -1 \leq t \leq 1 \qquad (i = 1,2, \ . \ . \ . \ , n)$$

with the initial conditions

$$\phi_i(-1) = \eta_i \qquad (i = 1,2, \ . \ . \ . \ ,n)$$

The approximate Chebyshev series solution and derivatives are provided in the form of the finite series

$$\phi_i^{(p)}(t) \approx \sum_{k=0}^{M+1-p}{}' \ B_{k,i}^{(p)} T_k(t) \qquad (p = 0,1; \ i = 1,2, \ . \ . \ . \ ,n)$$

where $T_k(t)$ are Chebyshev polynomials defined by

$$T_k(t) = \cos(k \cos^{-1} t) \ , \qquad -1 \leq t \leq 1$$

The accuracy of the approximate solution depends on the convergence error ε and the degree of polynomial approximation M prescribed by the user. When both $\varepsilon \to 0$ and $M \to \infty$, the approximate Chebyshev series solution approaches that of the infinite Chebyshev series expansion (see main body of the report for choice of ε and M and the estimation of errors).

Usage

The routine is entered via the statement

```
CALL ALG1 (N, KIN, M + 1, M + 2, ETA, EPSN, KIT, TR, PHI, XR, XS, B, IC,
          NER, DERIV)
```

where

N(=n) is the number of first-order differential equations.

KIN is an integer code used to indicate the method of computation desired:
KIN = 0, for straight iteration, (see algorithm I, page 13).
KIN = 1, for iteration with modification of columns (see page 26).
KIN = 2, for iteration with modification of rows (see page 25).

M is the degree of polynomial approximation used to represent the derivatives $\phi_i'(t) = \psi_i(t,\phi_1,\phi_2, \ldots ,\phi_n)$.

ETA is a double-precision array of 3n locations. The first n locations are used to store the n initial conditions η_i $(i = 1,2, \ldots ,n)$ in the order of ascending i. The remaining locations are used internally by the subroutine.

EPSN is the convergence error ε prescribed by the user and is a double-precision variable.

KIT is the maximum number of iterations allowed by the user.

TR is a double-precision array of M + 2 locations reserved as working storage for the subroutine.

PHI is a double-precision array reserved as working storage for the subroutine. The number of locations allocated are

2n(M + 2)	for	KIN = 0
n(n + 3)(M + 2)	for	KIN = 1
n(M + 4)(M + 2)	for	KIN = 2.

XR is a double-precision array of n(4n + 2) locations reserved exclusively as working storage for the case with KIN = 1. XR is a dummy double-precision variable for the cases with KIN = 0 and 2.

XS is a double-precision array of (M + 1)(4M + 6) locations reserved exclusively as working storage for the case with KIN = 2. XS is a dummy double-precision variable for the cases with KIN = 0 and 1.

B is a double-precision three-dimensional array of dimension $(M + 2) \times N \times 3$. The first two rectangular arrays of dimension $(M + 2) \times N$ are used to store the approximate Chebyshev coefficients of $\phi_i'(t)$ with

$$B_{k,i}^{(0)} \quad \text{stored in} \quad B(K + 1, I, 1)$$

for $i = I = 1, 2, \ldots, n$; $k = K = 0, 1, \ldots, M + 1$

$$B_{k,i}^{(1)} \quad \text{stored in} \quad B(K + 1, I, 2)$$

for $i = I = 1, 2, \ldots, n$; $k = K = 0, 1, \ldots, M$

The remaining locations are used internally by the subroutine.

IC is the number of iterations executed by the subroutine to achieve convergence.

NER is the error code. NER = 0 is a normal return; NER = 1 indicates that the number of iterations had exceeded KIT.

DERIV is the name of a user supplied subroutine for the computation of the n first derivatives $\phi_i'(t)$ (see Derivative Subroutine below). The name DERIV (or whatever name the user chooses) must appear in an EXTERNAL specification statement in the calling program.

Derivative Subroutine

A subroutine for the computation of the first derivatives $\phi_i'(t)$ $(i = 1, 2, \ldots, n)$ must be supplied by the user and must be of the following format:

```
SUBROUTINE DERIV (N, T, PHI, PSI)
DOUBLE PRECISION  T, PHI, PSI
DIMENSION  PHI(N), PSI(N)
PSI(1) = . . .
PSI(2) = . . .
.
.
.
PSI(N) = . . .
RETURN
END
```

The symbols of the DERIV subroutine are defined as follows:

N(=n) is the number of first-order differential equations.

T is the independent variable.

PHI is an array of n locations used to store the values of $\phi_i(t)$ $(i = 1, 2, \ldots, n)$ in the order of ascending i.

PSI is an array of n locations used to store the n first derivatives $\phi_i'(t) = \psi_i(t, \phi_1, \phi_2, \ldots, \phi_n)$ $(i = 1, 2, \ldots, n)$ in the order of ascending i.

Other Subroutines Required

The subroutines

1. AL CHBY
2. AL DPIN

Evaluation of Solution and Derivatives

The finite series

$$\phi_i^{(p)}(t) \approx \sum_{k=0}^{M+1-p}{}' B_{k,i}^{(p)} T_k(t) \qquad (p = 0,1;\ i = 1,2, \ldots ,n)$$

may be evaluated as a function of the independent variable t by means of subroutine AL CHBY (see page 59), which in this case may be accessed via the statement

```
CALL CHBY (B(1, I, P + 1), M + 2 - P, T, SUM)
```

where

I, M, P are integers set equal to i, M and p, respectively.

B(1,I,P + 1) is the location of $B_{i,0}^{(p)}$.

T is the independent variable t.

SUM is the evaluated result.

Subroutine AL ALG2

Identification

AL ALG2, Chebyshev Series Integration of an nth-order Nonlinear Differential Equation
FORTRAN IV, Double-Precision Subroutine

Purpose

This subroutine is used to generate an approximate Chebyshev series solution for an nth-order nonlinear differential equation with n initial conditions. The nth-order differential equation is of the form

$$\phi^{(n)}(t) = \psi\left(t, \phi, \phi', \ldots, \phi^{(n-1)}\right), \qquad -1 \le t \le 1$$

with the initial conditions

$$\phi^{(i)}(-1) = \eta_i \qquad (i = 0,1, \ldots ,n-1)$$

The approximate Chebyshev solution and its derivatives are provided in the form of the finite series

$$\phi^{(i)}(t) \approx \sum_{k=0}^{M+n-i} B_k^{(i)} T_k(t) \qquad (i = 0,1, \ldots ,n)$$

where $T_k(t)$ are Chebyshev polynomials defined by

$$T_k(t) = \cos(k \cos^{-1} t) , \qquad -1 \leq t \leq 1$$

The accuracy of the approximate solution depends on the convergence error ε and the degree of polynomial approximation M prescribed by the user. When both $\varepsilon \to 0$ and $M \to \infty$, the approximate Chebyshev series approaches that of the infinite Chebyshev series expansion. (See main body of the report for choice of ε and M and the estimation of errors.)

Usage

The routine is entered via the statement

```
CALL ALG2 (N, KIN, M + 1, M + 2, ETA, EPSN, KIT, TR, PHI, XR, B, IC, NER,
     NDER)
```

where

N(=n) is the order of the differential equation.

KIN is an integer code used to indicate the method of computation desired:
KIN = 0, for straight iteration (see algorithm II, page 30).
KIN = 1, for iteration with modification of individual entries (see page 34).
KIN = 2, for iteration with modification of rows (see page 34).

M is the degree of polynomial approximation used to represent the nth derivative $\phi^{(n)}(t) = \psi\left(t,\phi,\phi', \ldots ,\phi^{(n)}\right)$.

ETA is a double-precision array of 2n locations. The first n cells are used to store the n initial conditions η_i $(i = 0,1, \ldots ,n-1)$ in the order of ascending i. The remaining cells are used internally by the subroutine.

is the convergence error ε prescribed by the user and is a double-precision variable.

TR is a double-precision array of $M + 2$ cells reserved as working storage for the subroutine.

PHI is a double-precision array reserved as working storage for the subroutine. The number of locations allocated are

$$(n + 1)(M + 2) \quad \text{for} \quad KIN = 0$$
$$(n + 3)(M + 2) \quad \text{for} \quad KIN = 1$$
$$(M + 2)(M + n + 3) \quad \text{for} \quad KIN = 2.$$

XR is a double-precision array of $(M + 1)(4M + 6)$ locations reserved exclusively as working storage for the case with $KIN = 2$. XR is a dummy double-precision variable for the cases with $KIN = 0$ and 1.

B is a double-precision two-dimensional array of dimension $(M + n + 1) \times (n + 2)$ with

$$B_k^{(i)} \quad \text{stored in} \quad B(K + 1, I + 1)$$

for $i = I = 0, 1, \ldots, n$; $k = K = 0, 1, \ldots, M + n - i$. The remaining cells are used internally by the subroutine.

IC is the number of iterations executed by the subroutine to achieve convergence.

NER is the error code. NER = 0 is a normal return; NER = 1 indicates that the number of iterations exceeded KIT.

NDER is the name of a user supplied subroutine for the computation of the nth derivative $\phi^{(n)}(t)$ (see Derivative Subroutine below). The name NDER (or whatever name the user chooses) must appear in an EXTERNAL specification statement in the calling program.

Derivative Subroutine

A subroutine must be supplied by the user to compute the nth-derivative $\phi^{(n)}(t)$ and must be of the following format:

```
SUBROUTINE NDER (N, T, PHI, PSI)
DOUBLE PRECISION  T, PHI, PSI
DIMENSION  PHI(N)
PSI = . . .
RETURN
END
```

The symbols of the NDER subroutine are defined as follows:

N(=n) is the order of the differential equation.

T is the independent variable t.

PSI is the value of nth derivative $\phi^{(n)}(t) = \psi\left(t, \phi, \phi', \ldots, \phi^{(n-1)}\right)$

PHI is an array of n locations used to store the values of $\phi^{(i)}(t)$ $(i = 0, 1, \ldots, n - 1)$ in the order of ascending i.

Other Subroutines Required

1. AL CHBY
2. AL DPIN

Evaluation of Solution and Derivatives

The finite series

$$\phi^{(i)}(t) \approx \sum_{k=0}^{M+n-i}{}' B_k^{(i)} T_k(t) \qquad (i = 0,1, \ldots ,n)$$

may be evaluated as a function of the independent variable t by means of subroutine AL CHBY (see below), which in this case may be accessed via the statement

CALL CHBY (B(1, I + 1), M + 1 + N - I, T, SUM)

where

I, N, M are integers that take on the values of i, n, and M, respectively.

B(1,I+1) is the location of $B_0^{(i)}$.

T is the independent variable.

SUM is the evaluated result.

Subroutine AL CHBY

AL CHBY Evaluation of A Finite Series of Chebyshev Polynomials
FORTRAN IV, Double-Precision Subroutine

Purpose

This subroutine is used to evaluate a finite series of the form

$$S(t) = \sum_{k=0}^{N}{}' B_k T_k(t) \ , \qquad -1 \le t \le 1$$

where $T_k(t)$ are Chebyshev polynomials defined by

$$T_k(t) = \cos(k \cos^{-1} t)$$

Usage

This subroutine is entered via the statement

```
CALL CHBY (B, N + 1, T, SUM)
```

where

B is a double-precision array of $N + 1$ locations used to store B_k in the order of ascending k.

N is the order of the highest order Chebyshev polynomial.

T is the independent variable t and is a double-precision variable.

SUM is the sum $S(t)$ and a double-precision variable.

Method (See eq. (A23).)

Subroutine AL DPIN

Identification

AL DPIN, Matrix Inversion
FORTRAN IV, Double-Precision Subroutine

Purpose

This subroutine is used to calculate the inverse of a square matrix A.

Usage

This subroutine is entered via the statement

```
CALL DPIN (A, N, KDET)
```

The parameters are defined as follows:

A is a double-precision two-dimensional array of dimension $N \times N$ used to store elements of the matrix A. Upon return, the inverse A^{-1} will be found in the array A.

N is the order of matrix A.

KDET is an error code. KDET = 0 is a normal return; KDET = 1 indicates that A is singular.

Method

Jordan's method of elimination is used to calculate A^{-1} (see ref. 14).

```
$IBFTC ALALG1
C ALALG1, KIN L. LEE, APRIL 1969
C ALGORITHM I
C CHEBYSHEV SERIES INTEGRATION OF A SYSTEM OF
C N FIRST-ORDER NON-LINEAR DIFFERENTIAL EQUATIONS
      SUBROUTINE  ALG1(N,KIN,NC1,NC2,ETA,EPSN,KIT,T,PHI,XR,XS,B,IC,NER,
     1                 DERIV)
      DOUBLE PRECISION ONC1,T,OJ,PI,ETA,F2,Z,S,SIGN,CHK,EPSN,PHI,OK,B,
     1 XR,XS,DET
      DIMENSION T(1),PHI(NC2,N,1),ETA(N,1),B(NC2,N,1),XR(N,1),XS(NC1,1)
      DATA F2,PI
     1 /2.0D+00,+0.3141592653589793D+01/
      NC=NC1-1
      NC3=NC+3
      NC4=NC+4
      NC5=NC+5
      NP2=N+2
      NP3=N+3
      NP4=N+4
      ONC1=NC1
      IN=KIN+1
      T(1)=1.D+00
      T(NC2)=  -1.D+00
      OJ=0
      DO 12 J1=2,NC1
      OJ=OJ+1.D+00
   12 T(J1) =DCOS(OJ*PI/ONC1)
      CALL DERIV(N,T(NC2),ETA,ETA(1,3))
      DO 92 I=1,N
   92 PHI(NC2,I,2)=ETA(I,3)
      DO 31 I=1,N
      DO 31 K1=1,NC2
   31 B(K1,I,3)=0
      DO 250 I=1,N
  250 B(NC2,I,2)=0
      GO TO(33,37,35),IN
   37 NR=NP2
      GO TO 36
   35 NR=NC3
   36 DO 34 I=1,N
      DO 34 L=2,NR
   34 PHI(NC2,I,L+1)=PHI(NC2,I,2)
   33 DO 43 I=1,N
      DO 43 K1=1,NC1
   43 PHI(K1,I,1)=ETA(I,1)
      IC=1
      J=1
```

```
      GO TO 44
   40 DO 42 I=1,N
      DO 42 J1=1,NC1
   42 CALL CH BY(B(1,I,1),NC2,T(J1),PHI(J1,I,1))
   44 DO 52 J1=1,NC1
      DO 51 I=1,N
   51 ETA(I,2)=PHI(J1,I,1)
      CALL DERIV(N,T(J1),ETA(1,2),ETA(1,3))
      DO 52 I=1,N
   52 PHI(J1,I,J+1)=ETA(I,3)
   60 DO 63 I=1,N
      PHI(NC2,I,J+1)=PHI(NC2,I,J+1)/F2
      DO 62 K1=1,NC1
      CALL CH BY(PHI(1,I,J+1),NC2,T(K1),Z)
   62 B(K1,I,2)=F2*Z/ONC1
   63 PHI(NC2,I,J+1)=F2*PHI(NC2,I,J+1)
      DO 72 I=1,N
      OK=0
      DO 71 K1=2,NC1
      OK=OK+1.D+00
   71 B(K1,I,1   )
     1        =(B(K1-1,I,2)-B(K1+1,I,2))/(F2*OK)
   72 B(NC2,I,1)=B(NC1,I,2)/(F2*ONC1)
      DO 82 I=1,N
      S=0.
      SIGN=1.D+00
      DO 81 K1=2,NC2
      SIGN= -SIGN
   81 S=S+SIGN*B(K1,I,1)
   82 B(1,I,1)=F2*(ETA(I,1)-S)
  260 IF((J.EQ.1).AND.(KIN.NE.0))GO TO 113
      IF(IC.EQ.1)GO TO 131
  100 DO 102 I=1,N
      DO 102 K1=1,NC2
      CHK=DABS(B(K1,I,2)-B(K1,I,3))
      IF(CHK.GT.EPSN) GO TO 110
  102 CONTINUE
      NER   =0
      GO TO 151
  110 IF(IC.EQ.KIT) GO TO 120
      GO TO (131,111,112),IN
  111 IF(J.EQ.NP2)GO TO 180
      GO TO 113
  112 IF(J.EQ.NC3)GO TO 181
  113 J=J+1
  131 IC=IC+1
      DO 240 I=1,N
```

```
      DO 240 K1=1,NC2                                               ALG10094
  240 B(K1,I,3)=B(K1,I,2)                                           ALG10095
      GO TO 40                                                      ALG10096
  120 NER   =1                                                      ALG10097
      GO TO 151                                                     ALG10098
  180 IF(N.EQ.1)GO TO 360                                           ALG10099
      DO 309 K1=1,NC1                                               ALG10100
      DO 310 J=1,NP2                                                ALG10101
      DO 310 I=1,N                                                  ALG10102
  310 XR(I,J)=PHI(K1,I,J+1)                                         ALG10103
      DO 320 J=1,N                                                  ALG10104
      L1=2*N+J+2                                                    ALG10105
      DO 320 I=1,N                                                  ALG10106
  320 XR(I,L1)=XR(I,J+2)-F2*XR(I,J+1)+XR(I,J)                       ALG10107
      CALL DPIN(XR(1,2*N+3),N,KDET)                                 ALG10108
      IF(KDET.NE.0)GO TO 380                                        ALG10109
      DO 340 J=1,N                                                  ALG10110
      L2=N+J+2                                                      ALG10111
      DO 340 I=1,N                                                  ALG10112
  340 XR(I,L2)=XR(I,J+1)-XR(I,J)                                    ALG10113
      DO 350 J=1,N                                                  ALG10114
      L3=3*N+J+2                                                    ALG10115
      L4=2*N+J+2                                                    ALG10116
      DO 350 I=1,N                                                  ALG10117
      XR(I,L3)=0                                                    ALG10118
      DO 350 K=1,N                                                  ALG10119
      L5=N+K+2                                                      ALG10120
  350 XR(I,L3)=XR(I,L3)-XR(I,L5)*XR(K,L4)                           ALG10121
      DO 309 I=1,N                                                  ALG10122
      S=XR(I,N+1)                                                   ALG10123
      DO 390 K=1,N                                                  ALG10124
      L6=3*N+K+2                                                    ALG10125
  390 S=S+XR(I,L6)*(XR(K,N+2)-XR(K,N+1))                            ALG10126
  309 PHI(K1,I,2)=S                                                 ALG10127
      GO TO 182                                                     ALG10128
  380 DO 381 I=1,N                                                  ALG10129
      DO 381 K1=1,NC1                                               ALG10130
  381 PHI(K1,I,2)=PHI(K1,I,NP3)                                     ALG10131
  384 J=2                                                           ALG10132
      GO TO 131                                                     ALG10133
  360 DO 370 K1=1,NC1                                               ALG10134
      DET=PHI(K1,1,4)-F2*PHI(K1,1,3)+PHI(K1,1,2)                    ALG10135
      IF(DET.EQ.0.)GO TO 380                                        ALG10136
  370 PHI(K1,1,2)=PHI(K1,1,4)-((PHI(K1,1,4)-PHI(K1,1,3))**2)/DET    ALG10137
      GO TO 182                                                     ALG10138
  181 DO 409 I=1,N                                                  ALG10139
      DO 410 J=1,NC3                                                ALG10140
```

```
      DO 410 K1=1,NC1                                        ALG10141
  410 XS(K1,J)=PHI(K1,I,J+1)                                 ALG10142
      DO 420 J=1,NC1                                         ALG10143
      L1=2*NC+J+4                                            ALG10144
      DO 420 K1=1,NC1                                        ALG10145
  420 XS(K1,L1)=XS(K1,J+2)-F2*XS(K1,J+1)+XS(K1,J)            ALG10146
      CALL DPIN(XS(1,2*NC+5),NC1,KDET)                       ALG10147
      IF(KDET.NE.0)GO TO 382                                 ALG10148
      DO 440 J=1,NC1                                         ALG10149
      L2=NC+J+3                                              ALG10150
      DO 440 K1=1,NC1                                        ALG10151
  440 XS(K1,L2)=XS(K1,J+1)-XS(K1,J)                          ALG10152
      DO 450 J=1,NC1                                         ALG10153
      L3=3*NC+J+5                                            ALG10154
      L4=2*NC+J+4                                            ALG10155
      DO 450 K1=1,NC1                                        ALG10156
      XS(K1,L3)=0                                            ALG10157
      DO 450 K=1,NC1                                         ALG10158
      L5=NC+K+3                                              ALG10159
  450 XS(K1,L3)=XS(K1,L3)-XS(K1,L5)*XS(K,L4)                 ALG10160
      DO 409 K1=1,NC1                                        ALG10161
      S=XS(K1,NC2)                                           ALG10162
      DO 490 K=1,NC1                                         ALG10163
      L6=3*NC+K+5                                            ALG10164
  490 S=S+XS(K1,L6)*(XS(K,NC3)-XS(K,NC2))                    ALG10165
  409 PHI(K1,I,2)=S                                          ALG10166
  182 J=1                                                    ALG10167
      GO TO 60                                               ALG10168
  382 DO 383 I=1,N                                           ALG10169
      DO 383 K1=1,NC1                                        ALG10170
  383 PHI(K1,I,2)=PHI(K1,I,NC4)                              ALG10171
      GO TO 384                                              ALG10172
  151 RETURN                                                 ALG10173
      END                                                    ALG10174
```

```
$IBFTC ALALG2                                                            ALG20000
C ALALG2, KIN L. LEE, APRIL 1969                                         ALG20001
C ALGORITHM II                                                           ALG20002
C CHEBYSHEV SERIES INTEGRATION OF AN NTH-ORDER NON-LINEAR                ALG20003
C DIFFERENTIAL EQUATION                                                  ALG20004
      SUBROUTINE ALG2(N,KIN,M1,M2,L,ETA,EPSN,KIT,T,PHI,XR,A,IC,NER,      ALG20005
     1                NDER)                                              ALG20006
      DOUBLE PRECISION ETA,A,T,PHI,F1,F2,F4,PI,F3,OM,OMP1,OM4,OJ,XR,     ALG20007
     1DF,OK,SIGN,S,OKP,Z,CHK,EPSN,DET                                    ALG20008
      DIMENSION ETA(N,1),A(L,1),T(1),PHI(M2,1),XR(M1,1)                  ALG20009
      DATA F1,F2,F4,PI,F3                                                ALG20010
     1/1.D+00,                                                           ALG20011
     2 2.D+00,                                                           ALG20012
     3 4.D+00,                                                           ALG20013
     4 0.3141592653589793D+01,                                           ALG20014
     5 3.D+00/                                                           ALG20015
      M=M2-2                                                             ALG20016
      NP1=N+1                                                            ALG20017
      NP2=N+2                                                            ALG20018
      NP3=N+3                                                            ALG20019
      NM1=N-1                                                            ALG20020
      MP3=M+3                                                            ALG20021
      MP5=M+5                                                            ALG20022
      IN=KIN+1                                                           ALG20023
      NPMP3=N+M+3                                                        ALG20024
      OM=M                                                               ALG20025
      OMP1=OM+F1                                                         ALG20026
      M4=M+N                                                             ALG20027
      OM4=M4                                                             ALG20028
      T(1)=F1                                                            ALG20029
      LP=L+1                                                             ALG20030
      T(M2)=-F1                                                          ALG20031
      OJ=0                                                               ALG20032
      DO 31 J1=2,M1                                                      ALG20033
      OJ=OJ+F1                                                           ALG20034
   31 T(J1 )=DCOS(OJ*PI/OMP1)                                            ALG20035
      DO 278 I1=1,N                                                      ALG20036
  278 PHI(M2,I1)=ETA(I1,1)                                               ALG20037
      CALL NDER(N,T(M2),ETA,DF)                                          ALG20038
      PHI(M2,NP1)=DF                                                     ALG20039
      GO TO(280,281,282),IN                                              ALG20040
  281 DO 284 J=NP2,NP3                                                   ALG20041
  284 PHI(M2,J)=DF                                                       ALG20042
      GO TO 280                                                          ALG20043
  282 DO 201 J=NP2,NPMP3                                                 ALG20044
  201 PHI(M2,J)=DF                                                       ALG20045
  280 DO 275 I1=1,N                                                      ALG20046
```

```
275 A(1,I1)=F2*ETA(I1,1)                                         ALG20047
    DO 276 I1=1,N                                                ALG20048
    DO 276 K1=2,L                                                ALG20049
276 A(K1,I1)=0.                                                  ALG20050
    DO 277 K1=1,L                                                ALG20051
277 A(K1,NP1)=0.                                                 ALG20052
    IC=1                                                         ALG20053
    J=1                                                          ALG20054
    DO 274 I1=1,N                                                ALG20055
    DO 274 K1=1,M2                                               ALG20056
274 PHI(K1,I1)=ETA(I1,1)                                         ALG20057
    GO TO 233                                                    ALG20058
220 DO 272 K1=1,M1                                               ALG20059
272 A(K1,NP2)=A(K1,NP1)                                          ALG20060
283 DO 221 I1=1,N                                                ALG20061
    LMI=LP-I1                                                    ALG20062
    DO 221 K1=1,M1                                               ALG20063
221 CALL CHBY(A(1,I1),LMI,T(K1),PHI(K1,I1))                      ALG20064
233 NPJ=N+J                                                      ALG20065
230 DO 232 K1=1,M1                                               ALG20066
    DO 234 I1=1,N                                                ALG20067
234 ETA(I1,2)=PHI(K1,I1)                                         ALG20068
232 CALL NDER(N,T(K1),ETA(1,2),PHI(K1,NPJ))                      ALG20069
240 PHI(M2,NPJ)=PHI(M2,NPJ)/F2                                   ALG20070
    DO 241 K1=1,M1                                               ALG20071
    CALL CHBY(PHI(1,NPJ),M2,T(K1),Z)                             ALG20072
241 A(K1,NP1)=F2*Z/(OMP1)                                        ALG20073
    PHI(M2,NPJ)=F2*PHI(M2,NPJ)                                   ALG20074
250 KP=NM1                                                       ALG20075
    OKP=KP                                                       ALG20076
251 M6=M4-KP                                                     ALG20077
    M7=M6-1                                                      ALG20078
    A(M6+1,KP+1)=A(M6,KP+2)/(F2*(OM4-OKP))                       ALG20079
    K=1                                                          ALG20080
    OK=K                                                         ALG20081
252 A(K+1,KP+1)=(A(K,KP+2)-A(K+2,KP+2))/(F2*OK)                  ALG20082
    IF(K.EQ.M7)GO TO 253                                         ALG20083
    K=K+1                                                        ALG20084
    OK=OK+F1                                                     ALG20085
    GO TO 252                                                    ALG20086
253 S=0                                                          ALG20087
    SIGN= F1                                                     ALG20088
    DO 254 K=1,M6                                                ALG20089
    SIGN=-SIGN                                                   ALG20090
254 S=S+SIGN*A(K+1,KP+1)                                         ALG20091
    A(1,KP+1)=F2*(ETA(KP+1,1)-S)                                 ALG20092
    IF(KP.EQ.0)GO TO 260                                         ALG20093
```

```
      KP=KP-1                                                           ALG20094
      OKP=OKP-F1                                                        ALG20095
      GO TO 251                                                         ALG20096
  260 IF((J.EQ.1).AND.(KIN.NE.0))GO TO 273                              ALG20097
      IF(IC.EQ.1)GO TO 330                                              ALG20098
      DO 261 K1=1,M1                                                    ALG20099
      CHK=DABS(A(K1,NP1)-A(K1,NP2))                                     ALG20100
      IF(CHK.GE.EPSN)GO TO 270                                          ALG20101
  261 CONTINUE                                                          ALG20102
      NER=0                                                             ALG20103
      GO TO 262                                                         ALG20104
  270 IF(IC.GE.KIT)GO TO 271                                            ALG20105
      GO TO(330,331,332),IN                                             ALG20106
  331 IF(J.EQ.3)GO TO 300                                               ALG20107
      GO TO 273                                                         ALG20108
  332 IF(J.EQ.MP3)GO TO 304                                             ALG20109
  273 J=J+1                                                             ALG20110
  330 IC=IC+1                                                           ALG20111
      GO TO 220                                                         ALG20112
  300 DO 301 K1=1,M1                                                    ALG20113
      DET=PHI(K1,NP3)-F2*PHI(K1,NP2)+PHI(K1,NP1)                        ALG20114
      IF(DET.EQ.0.)GO TO 302                                            ALG20115
  301 PHI(K1,NP1)=PHI(K1,NP3)-((PHI(K1,NP3)-PHI(K1,NP2))**2)/DET        ALG20116
      GO TO 310                                                         ALG20117
  302 DO 305 K1=1,M1                                                    ALG20118
  305 PHI(K1,NP1)=PHI(N1,NP3)                                           ALG20119
      GO TO 306                                                         ALG20120
  304 CONTINUE                                                          ALG20121
      DO 303 J=1,MP3                                                    ALG20122
      NPJ=N+J                                                           ALG20123
      DO 303 K1=1,M1                                                    ALG20124
  303 XR(K1,J)=PHI(K1,NPJ)                                              ALG20125
      DO 420 J=1,M1                                                     ALG20126
      L1=2*M+J+4                                                        ALG20127
      DO 420 K1=1,M1                                                    ALG20128
  420 XR(K1,L1)=XR(K1,J+2)-F2*XR(K1,J+1)+XR(K1,J)                       ALG20129
      CALL DPIN(XR(1,2*M+5),M1,KDET)                                    ALG20130
      IF(KDET.NE.0)GO TO 350                                            ALG20131
      DO 440 J=1,M1                                                     ALG20132
      L2=M+J+3                                                          ALG20133
      DO 440 K1=1,M1                                                    ALG20134
  440 XR(K1,L2)=XR(K1,J+1)-XR(K1,J)                                     ALG20135
      DO 450 J=1,M1                                                     ALG20136
      L3=3*M+J+5                                                        ALG20137
      L4=2*M+J+4                                                        ALG20138
      DO 450 K1=1,M1                                                    ALG20139
      XR(K1,L3)=0                                                       ALG20140
```

```
      DO 450 K=1,M1                                                     ALG20141
      L5=M+K+3                                                          ALG20142
  450 XR(K1,L3)=XR(K1,L3)-XR(K1,L5)*XR(K,L4)                            ALG20143
      DO 409 K1=1,M1                                                    ALG20144
      S=XR(K1,M2)                                                       ALG20145
      DO 410 K=1,M1                                                     ALG20146
      L6=3*M+5+K                                                        ALG20147
  410 S=S+XR(K1,L6)*(XR(K,MP3)-XR(K,M2))                                ALG20148
  409 PHI(K1,NP1)=S                                                     ALG20149
  310 J=1                                                               ALG20150
      NPJ=N+J                                                           ALG20151
      GO TO 240                                                         ALG20152
  350 DO 351 K1=1,M1                                                    ALG20153
  351 PHI(K1,NP1)=PHI(K1,NPMP3)                                         ALG20154
  306 J=2                                                               ALG20155
      GO TO 330                                                         ALG20156
  271 NER=1                                                             ALG20157
  262 RETURN                                                            ALG20158
      END                                                               ALG20159
```

160

```
$IBFTC ALCHBY                                                          CHBY0000
C ALCHBY, KIN L. LEE, APRIL 1969                                       CHBY0001
C SUBROUTINE TO EVALUATE A FINITE SERIES OF CHEBYSHEV POLYNOMIALS      CHBY0002
      SUBROUTINE CHBY(A,NP1,T,PHI)                                     CHBY0003
      DOUBLE PRECISION A,T,PHI,CKP2,CKP1,CK                            CHBY0004
      DIMENSION A(NP1)                                                 CHBY0005
      IF(NP1-2)14,15,16                                                CHBY0006
   14 PHI=A(1)/2.D+00                                                  CHBY0007
      GO TO 17                                                         CHBY0008
   15 PHI=A(1)/2.D+00+A(2)*T                                           CHBY0009
      GO TO 17                                                         CHBY0010
   16 IF(NP1.GT.3) GO TO 10                                            CHBY0011
      CKP1=A(NP1)                                                      CHBY0012
      CK=A(NP1-1)+2.D+00*T*CKP1                                        CHBY0013
      GO TO 13                                                         CHBY0014
   10 CKP2=A(NP1)                                                      CHBY0015
      CKP1=A(NP1-1)+2.D+00*T*CKP2                                      CHBY0016
      K=NP1-2                                                          CHBY0017
   12 CK=A(K)+(2.D+00*T*CKP1-CKP2)                                     CHBY0018
      IF(K.EQ.2) GO TO 13                                              CHBY0019
      K=K-1                                                            CHBY0020
      CKP2=CKP1                                                        CHBY0021
      CKP1=CK                                                          CHBY0022
      GO TO 12                                                         CHBY0023
   13 PHI=A(1)/2.D+00+(CK*T-CKP1)                                      CHBY0024
   17 RETURN                                                           CHBY0025
      END                                                              CHBY0026
```

27

```
$IBFTC ALDPIN
C ALDPIN, KIN L. LEE, APRIL 1969
C MATRIX INVERSION BY JORDAN ELIMINATION
      SUBROUTINE DPIN(A,N,KDET)
      DIMENSION IPIVOT(100),INDEX(100,2),A(N,N)
      DOUBLE PRECISION AMAX,A,TEMP
      DO 20 J=1,N
   20 IPIVOT(J)=0
      DO 560 I=1,N
      AMAX=0
      DO 105 J=1,N
      IF(IPIVOT(J).GE.1)GO TO 105
      DO 100 K=1,N
      IF(IPIVOT(K).GE.1)GO TO 100
      TEMP=DABS(A(J,K))
      IF(AMAX.GE.TEMP)GO TO 100
      IROW=J
      ICOLUM=K
      AMAX=TEMP
  100 CONTINUE
  105 CONTINUE
      IF(AMAX.EQ.0.0)GO TO 760
      IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
      IF(IROW.EQ.ICOLUM)GO TO 260
      DO 200 L=1,N
      TEMP=A(IROW,L)
      A(IROW,L)=A(ICOLUM,L)
  200 A(ICOLUM,L)=TEMP
  260 INDEX(I,1)=IROW
      INDEX(I,2)=ICOLUM
      TEMP=A(ICOLUM,ICOLUM)
      A(ICOLUM,ICOLUM)=1.0
      DO 350 L=1,N
  350 A(ICOLUM,L)=A(ICOLUM,L)/TEMP
      DO 550 L1=1,N
      IF(L1.EQ.ICOLUM)GO TO 550
      TEMP=A(L1,ICOLUM)
      A(L1,ICOLUM)=0.0
      DO 450 L=1,N
  450 A(L1,L)=A(L1,L)-A(ICOLUM,L)*TEMP
  550 CONTINUE
  560 CONTINUE
      DO 710 I=1,N
      L=N+1-I
      IF(INDEX(L,1).EQ.INDEX(L,2))GO TO 710
      IROW=INDEX(L,1)
      ICOLUM=INDEX(L,2)
```

```
      DO 705 K=1,N                                                      DPIN0047
      TEMP=A(K,IROW)                                                    DPIN0048
      A(K,IROW)=A(K,ICOLUM)                                             DPIN0049
      A(K,ICOLUM)=TEMP                                                  DPIN0050
  705 CONTINUE                                                          DPIN0051
  710 CONTINUE                                                          DPIN0052
      KDET=0                                                            DPIN0053
      GO TO 740                                                         DPIN0054
  760 KDET=1                                                            DPIN0055
  740 RETURN                                                            DPIN0056
      END                                                               DPIN0057
```

REFERENCES

1. Clenshaw, C. W.: The Numerical Solution of Linear Differential Equations in Chebyshev Series. Proc. Cambridge Phil. Soc., vol. 53, 1957, pp. 134-149.

2. Elliott, David: A Chebyshev Series Method for the Numerical Solution of Fredholm Integral Equations. Computer J., vol. 6, no. 1, April 1963, pp. 102-111.

3. Clenshaw, C. W.; and Norton, H. J.: The Solution of Nonlinear Ordinary Differential Equations in Chebyshev Series. Computer J., vol. 6, no. 1, April 1963, pp. 88-92.

4. Norton, H. J.: The Iterative Solution of Nonlinear Ordinary Differential Equations in Chebyshev Series. Computer J., vol. 7, no. 1, April 1964, pp. 76-85.

5. Fraser, W.; and Hart, J. F.: Near-Minimax Polynomial Approximation and Partitioning of Intervals. Communications of ACM, vol. 7, no. 8, Aug. 1964, pp. 486-489.

6. Abramowitz, M.; and Stegun, I. A., eds.: Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables. National Bureau of Standards, U. S. Dept. of Commerce, 1964, p. 200.

7. Tenenbaum, M.; and Pollard, H.: Ordinary Differential Equations. Harper and Row, N. Y., 1963.

8. Fox, L.: Numerical Solution of Ordinary and Partial Differential Equations. Addison-Wesley Publishing Co., Reading, Mass., 1962.

9. Henrici, Peter: Elements of Numerical Analysis. John Wiley and Sons, 1964.

10. Rivlin, T. J.: Čebyšev Expansions and Best Uniform Approximation. IBM Research Rep. RZ-93, 1962.

11. Lanczos, C.: Tables of Chebyshev Polynomials $S_n(x)$ and $C_n(x)$. National Bureau Standards, U. S. Dept. of Commerce, Applied Mathematics Series 9, Dec. 1952.

12. Clenshaw, C. W.: A Note on the Summation of Chebyshev Series. Mathematical Tables and Other Aids to Computations, vol. 9, no. 51, July 1955, pp. 118-120.

13. Natanson, I. P.: Constructive Theory of Functions (Book I), AEC-tr-4503 translated from Russian by Office of Technical Services, U. S. Dept. of Commerce, 1961.

14. Fox, L.: An Introduction to Numerical Linear Algebra with Exercises. Oxford University Press, N. Y., 1965.

NASA-Langley, 1969 —— 19 A-3302

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D. C. 20546

OFFICIAL BUSINESS

FIRST CLASS MAIL

POSTAGE AND FEES PAID
NATIONAL AERONAUTICS AND
SPACE ADMINISTRATION

POSTMASTER: If Undeliverable (Section 158 Postal Manual) Do Not Return

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Notes, and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546